Guidelines for forestry information processing

with particular reference to developing countries

Based on the work of
J.W. van Roessel

The designations employed and the presentation of material in this publication do not imply the expression of any opinion whatsoever on the part of the Food and Agriculture Organization of the United Nations concerning the legal status of any country, territory, city or area or of its authorities, or concerning the delimitation of its frontiers or boundaries.

The mention of specific companies or of their products or brand names does not imply any endorsement or recommendation on the part of the Food and Agriculture Organization of the United Nations.
It is further pointed out that the survey of the products is limited mainly to systems developed in North America.

M-35

ISBN 92-5-102451-0

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying or otherwise, without the prior permission of the copyright owner. Applications for such permission, with a statement of the purpose and extent of the reproduction, should be addressed to the Director, Publications Division, Food and Agriculture Organization of the United Nations, Via delle Terme di Caracalla, 00100 Rome, Italy.

© FAO 1986

PREFACE

The transfer of technology and promotion of self-reliance are among the main objectives of FAO's technical assistance to developing countries. As part of this mandate, the Forestry Department of FAO published in 1973 a manual on forest inventory, with special reference to the tropics, and in 1979 a generalized computer software package to facilitate computerization of the forest inventories data processing (FIDAPS).

Keeping in view the increasing applications of computer technology in forestry, the Department undertook preparation of this report which deals with a wider area than forest inventory and includes a detailed presentation of geographic information systems and guidelines for the choice of hardware and software, with particular reference to developing countries.

The services of Dr. Jan van Roessel, presently working at the EROS Data Centre of the National Mapping Division, U.S. Geological Survey, were used to make a review of the state of the art and to carry out case studies of computer systems (including hardware and software) installed at selected forestry institutions of developing countries, viz. Brazil, Burma and India. He worked in close cooperation with Dr. K.D. Singh, Senior Forestry Officer of the Forestry Department, responsible for this element of work.

It is hoped that this report will provide useful guidance to those in the forestry field who are designing computer applications in developing countries, and, in doing so, promote self-reliance in this area.

M.A. Flores Rodas
Assistant Director-General
Forestry Department

TABLE OF CONTENTS

1 INTRODUCTION

In the last decades there has been an unparalleled accelerating evolution of computing devices. This development is profoundly influencing all aspects of life, including resource management disciplines such as forestry.

Quite recently the influence of computers in the practice of forestry has taken another quantum jump with the availability of adequate computer power to construct information systems that can deal with spatial data. Whereas computers have been traditionally used to process statistical and other data, now they are powerful enough to keep track of maps and related statistics for day to day resource management and long term planning. In addition there has been the very recent development of the microcomputer, resulting in a real breakthrough in the affordability of computer power and data storage.

As forestry is often practiced conservatively, these developments have had a tendency to lag behind those in other disciplines and professions. This gap is probably even more pronounced when considering the present status in developing countries.

However the potential benefits that may result from the use of better and more timely information through the use of computing machinery may be enormous, and therefore every effort must be made to close the gap. In doing so however, there are many potential pitfalls and an array of bewildering options, especially for developing countries.

To provide some guidance to the explorer of computerization possibilities, to avoid these pitfalls and to help him select the best options for his situation, the Food and Agriculture Organization of the UN has prepared this report.

1.1 Objectives

The overall objective of this report is to provide guidelines for forestry data processing in developing countries. This main objective can be broken down into several more specific goals.

To make intelligent decisions in developing data processing capabilities, one should have an awareness of the current status of available technology. Therefore, chapter 2, Current Status of Technology, is entirely dedicated to an overall description and categorization of the present situation. This chapter is the main body of the report, simply because of the bewildering variety of ever expanding software and hardware that may be applicable to forestry management and planning problems. It has two main parts. The first part will discuss general aspects of computer technology, in terms of hardware, systems software and applications software systems, such as database management, statistical, and geographic information systems. The second part of the chapter is intended to deal with existing forestry applications, systems and packages, as divided into four main categories: inventory data processing,

management information systems, management planning systems, and economic and sector planning systems. For each of these categories there will be an important division between applications and systems for larger minicomputers and those designed for micro-systems.

Further, one should be aware of the problems that may exist or can be caused by applying the new technology in various situations in developing countries. Therefore, chapter 3 will attempt to address these aspects. Solutions and guidelines are presented in chapter 4. Chapter 5 of the report will present some case descriptions.

1.2 Scope And Limitations

The objectives in the previous section are the long term objectives for this report. There are several factors that make the development of the report as stated in the objectives and enormous undertaking, one that is perhaps too ambituous.

The technology involved is changing so rapidly (approximately a 100 fold increase in power every 10 years) that the report certainly would be outdated when fully developed. If this factor is combined with the enormously broad area of application of the technology in forestry, then one quickly becomes convinced of the enormity of the problem. However one cannot throw up one's hands, because the only way in which computerization can be properly applied is by being well informed. An attempt therefore has to be made even though incomplete.

A further restricting factor in relation to developing countries is that the report should apply to many different places and situations, which in in their full scope cannot be fully appreciated by one person. As it stands, it is hoped that the glimpse of the development situation obtained in the authors brief visits has helped to insert some reality in the report, but the technical guidelines are otherwise based on the assumption that the problems in correctly applying computer technology are basically the same, regardless of location. They may differ in severity.

The report has therefore been developed under the assumption that it will be very incomplete initially, but that over time, as more people become involved and periodic revisions and updates are made, it will become more representative.

The chapter on Forestry Applied Technology is currently very incomplete. A full table of contents in which all the intended sub-sections for this report are included is shown in Appendix C. The current version of the report has much material on geographic information systems, because this is the authors main area of expertise.

2 CURRENT STATUS OF COMPUTER TECHNOLOGY FOR FORESTRY INFORMATION PROCESSING

2.1 General Technology

Current computer technology is subject to such an explosive growth that the current state-of-the-art almost defies description. What is in use today may totally obsolete tomorrow. But, a user buying hardware essentially freezes the level of technology for his particular application and his equipment becomes obsolete only when it no longer serves his needs and requirements. Thus, the descriptions in this report maybe applicable for some time, although at the time of publication there probably will be new generations of equipment that can outperform earlier state-of-the-art machinery and software. However, especially where hardware is concerned, it will be more profitable to focus on trends rather than on system descriptions.

The division into hardware and software of the next two sections has been much used in the past, and for reasons of tradition is used in this report also, even though a more useful distinction can be made between the 'host environment' and the applications software. The host environment consists of hardware, operating system, programming languages and utilities, and is the substrate for the application packages. Keeping this in mind, we will briefly touch on the software aspect of the host environment at the beginning of the software section, but dedicate the larger part of this section to forestry applicable general applications software, primarily database management, geographic information systems and statistical systems.

2.1.1 Hardware

Hardware technology is currently developing at such a breakneck pace that it has become very hard for users and vendors to keep abreast of developments. The use of the technology is also snowballing, increasing by as much as 25% each year.

Because of the rapidly changing technology a prospective system buyer should be aware of current hardware trends. We will therefore devote this section to a description of these perceived trends. Much of the reporting is based on current developments in the U.S.. Our thesis is that the trends are more or less universal, affecting different places at different times because of economic and national policy considerations.

The trends can be summarized as follows:

1. General computer hardware is becoming more powerful, while more capacity can be obtained at decreasing cost.

2. Complexity of applications is rapidly increasing towards the micro end of the traditional mainframe-mini-micro line.

3. The role of mainframe computers is changing: they are more and more

becoming 'file serving' machines, maintaining databases that are otherwise too large for the micros and minis.

4. With mini computers becoming as powerful as the older mainframe machines, the mainframe classification is becoming obsolete, being replaced by a class of 'supercomputers'.

5. Computers are increasingly organized into interconnecting networks.

6. Applications that formerly did not comfortably fit on smaller machines are being transplanted to larger more powerful computers.

7. The emphasis for microcomputers is shifting from 8 to 16 bit processors and in the future will be on 32 bit processors.

8. Mini and mainframe manufacturers are adding models to both the top and bottom of their lines.

9. Manufacturers are increasingly integrating system architecture from the top to the bottom of the line.

10. Disk storage is becoming increasingly cheaper and denser (disk storage is one of the fastest evolving peripheral items).

11. There is an increasing emphasis on graphics.

12. Workstations with multiple windows and spatial inputs (mouse or digitizing tablet) are becoming more prevalent.

13. Increased competitions will force many microcomputer vendors to go out of business.

14. Cheaper multi-user computers will become more prevalent.

15. Some software functions will be implemented in hardware, giving rise to such peripherals as the back-end database machine.

Some explanatory comments, examples and references are provided in the following:

- *More powerful hardware.* General computer hardware is becoming more powerful, while more capacity can be obtained at decreasing cost. In terms of processing there is a tendency to incorporate the newer powerful 32 bit microprocessors in new microcomputers. The race for a 256 Kilobyte chip is over, American Telephone and Telegraph employs this chip in their 3B5 computer (this firm is expected to become number two in microcomputers in the U.S.A. by 1988). Several fims are now developing a 1 Megabyte chip. Disk storage technology is rapidly changing, providing more capacity for similar

costs.

Portable computers are improving rapidly: the newer machines weighing less than 10 pounds as opposed to the heavier and bulkier models from a few years ago.

Prices in general are still coming down, because the computer market is getting very competitive.There appears to be a floor on the price of peripherals with mechanical parts however. In the future prices may increase again because of the tremendous cost of marketing and support, especially in the microcomputer area.

o *Increasing complexity of applications.* Complexity of applications is rapidly increasing towards the micro end of the traditional mainframe-mini-micro line. For example, considering business decision support software (wordprocessing, spreadsheet, database), it is anticipated that there will be little difference between mainframe and microcomputer software by the end of the year. People spending $10,000/month on a service bureau can now move to a microcomputer, although most microcomputer packages will not offer the most sophisticated types of analysis.

o *Changing role of mainframes.* The role of mainframe computers is changing: they are more and more becoming 'file serving' machines, maintaining databases that are otherwise too large for the micros and minis.

o *Minis replacing mainframes.* With mini computers becoming as powerful as the older mainframe machines, the mainframe classification is becoming obsolete, being replaced by a class of 'supercomputers'.

o *Networks.* Computers are increasingly organized into interconnecting networks. People are beginning to see a need for linking personal computers and sharing resources such as printers and files. There is need for distributed database management systems. This means a distributed operating system, that can accomodate a number of networked computers so that the database management over the network is transparent to the user. This not yet happening, and will not happen for a while, but the need exists. A more down to earth configuration is for a smaller personal computer to communicate with a larger mini or mainframe. Thus, for instance, if one wants to perform financial planning on a microcomputer, but needs mainframe data one will be able to acces financial and operational information on the central mainframe database and download detailed information to the microcomputer. Accessibility to mainframe and larger minicomputers will drastically improve in the coming years. It is expected that personal computer local area networks will triple the size of their markets in the next two years.

o *More room for applications.* Applications that formerldid not comfortably fit on smaller machines are being transplanted to larger more powerful

computers.

- *32 bit microprocessors.* The emphasis for microcomputers is shifting from 8 to 16 bit processors and in the future will be on 32 bit processors. This does not mean that the different type processors will not stay around. People will begin to ask whether a machine will solve the problem, rather than what processor type it is. However 16 bit systems are of value because of the greater RAM size, more than 8 bit systems can provide. For smaller programming problems and word processing the 8 bit processor will stay popular. Thirty-two bit processors will be of great value for multi-user machines. They will take the place of the DEC PDP-11/35 or 11/45 minicomputer consisting of boards connected to a backplane via a bus. Already some computers exist in the $30,000-40,000 price range that are competing with $100,000-300,000 minicomputers. This will affect the pricing of many minicomputers.

- *New models at top and bottom of lines.* Mini and mainframe manufacturers are adding models to both the top and bottom of their lines. For instance, Digital Equipment Corporation (DEC) has recently announced its VAX 8600, which offers more than four times the processing power of the former top of the line VAX 11-785. There will also be a substantial expansion of the lower end of the line Professional 300 series. They are currently rewriting PDP-11 software, so that it will work on the Professional series, under a PDP line operating system.

  All minicomputer manufacturers are being pressured at the top and bottom: Perkin Elmer is developing high performance minicomputers, 7 mips and above, Data General is marketing a MicroEclipse.

- *Integrated architecture.* Manufacturers are increasingly integrating system architecture from the top to the bottom of the line. The idea is to develop a line of machines that is upward compatible from bottom to top. Prime Computer was one of the first companies to develop a line of compatible computers ranging from a model 450 to a model 850. A recent additon at the low end is the model 2250 (Rabbit) and at the high end the model 950. Data General has added a MicroEclipse, and DEC is marketing the MicroVAX. DEC has chosen not to continue its System 20 product line, but they are instead putting their energy behind the VAX line. They are pursuing the idea of a desktop VAX tied to a larger mainframe VAX with bigger disk drives. This change will make the product line coherent, rather than having a number of unrelated products such as PDP 8's, 11's, the VAX, the Professionals, the 10's and the 20's. It is anticipated that the PDP 11 line will be phased out, and that the VAX price will become low enough the make up the difference. Data General and Hewlett Packard are moving in the same direction. IBM now has a desktop 370. Minicomputers vendors are forced into this development, because if they do not provide the power at the low end of the line, some other manufacturer will. American Telephone and Telegraph will soon provide a 5 mips chip 32 bit processor, more than the current 1 mips per chip

of National Semiconductor. In all it is estimated that three times as many mips were sold on microcomputers as on minis and mainframes combined. Therefore, the minicomputer and mainframe concerns can only be expected to be hurt by the semiconductor leaders in the war for low cost mips.

- *Cheaper disk storage.* Disk storage is becoming increasingly cheaper and denser (disk storage is one of the fastest evolving peripheral items). In the microcomputer area, the market for floppy disk drives appears to be healthy and active. An increasing number of companies is producing microfloppies, of a size less than 5.25 inches. Only a year ago there were two major products in this area: the Hitachi 3 inch and Sony 3.5 inch disks. Now, three other companies are offering 3.25 inch disks: Tabor, MPI, and Seagate. Tandon, Shugart, Mitsubishi and Sony are leading producers of 3.5 inch floppies, which like the 3.25 inch products are now also available in formats that are plug compatible with the industry standard 5.25 inch disk. The smallest products are 3 inch disks (Hitachi, Maxell, TDK Electronics and Amdek). Trends are for 3.5 inch drives. Hewlett Packard claims to have shipped 50,000 3.5 inch drives, as part of the HP-150. This type of drive is more economical than 3 inch drives and requires less power than 5.25 inch drives. One megabyte drives in the 3.5 to 3.25 floppy format are already available. In 1982 5.25 inch drives surpassed 8 inch drives as the most popular formats, although 8 inch drives remain of major importance. A new development in this area is the 'half height' drive.

  A tremendous growth area for hard disk storage is found in the so called 'wini' (mini Winchester) area of disks less than 3.5 inches. A definite market exists for winis as a competition for floppies. Compaq computers has installed one with a 10 Megabyte capacity in its portable machine. It is anticipated that the the number of high capacity Winchester vendors will shrink to half a dozen. Price wars coupled with breakthroughs are making the 5-10 Megabayte Winchester more affordable. There has been a price drop of 15% in the high capacity (20 Megabyte and up) disks, and a drop of 20-25% in lower capacity disks (5-10 Megabyte). More complex operating systems and integrated applications software demand more disk capacity, and therefore benefit by this trend. A 5 Megabyte disk which holds a 4500 page novel once seemed large, but for current software it is barely sufficient. Winchester disks can seem small because they are difficult to recycle, a new solution to this problem is the removable Winchester disk cartridge. Some vendors use a tape cartridge as a back-up medium.

- *More graphics.* There is an increasing emphasis on graphics. As integrated chips become available more and more sophisticated high resolution graphics will be used in personal computers.

- *Workstations.* Workstations with multiple windows and spatial inputs (mouse or digitizing tablet) are becoming more prevalent.

- *More competition.* Increased competitions will force many microcomputer

vendors to go out of business. Many small computer makers are currently in trouble. It is anticipated that there will be two types of manufacturers left: those addressing the broad market areas such as IBM, Data General, Digital Equipment Corporation, Apple, Hewlett Pakcard. The second group will be developing special products for particular markets. They may expand out of those markets, but they will not be able to compete with the big companies.

- *Cheaper multi-user machines.* Cheaper multi-user computers will become more prevalent.

- *More hardwired functions.* Some software functions wilbe implemented in hardware, giving rise to such peripherals as the back-end database machine. This is a special hardware device solely dedicated to maintaining and serving data to the host machine or machines. Almost every corporation as well as many hardware and software vendors are working toward file server products. Many products on the market are still embryonic, but some, such as the Britton Lee intelligent database machine are currently in practical use, providing dependable service.

2.1.2 Software

2.1.2.1 Systems Software

Basic systems software need for the support of the user and applications environment can be assigned to four categories: 1) operating systems; 2) programming languages; 3) graphics support; 4) general utilities. In this section we will briefly discuss some trends and recent developments in each of these areas. An excellent recent review of the operating principles of the software in the above categories can be found in a September 1984 issue of Scientific American on software.

Operating systems on mini and larger computers can be assigned to a few classes that may be important to a prospective user. One distinction is between systems that support virtual memory and those that do not. With a virtual memory system the user has seemingly unlimited memory access. When physical memory is exhausted, the operating system automatically swaps 'pages' of data between memory and the disk, so that the memory seems limited only by the capacity of the disk. With a virtual memory a FORTRAN programmer can easily assign a few 1000x1000 arrays without problems. The only caution that should be exercised is to index each array such that excessive paging is avoided. A simple timing test in which a large two-dimensional array is traversed in I,J order and J,I order will reveal the problem. Depending on the order of indexing the time difference factor may be 100 times or greater. The newer operating systems will support virtual memory, and this is a great boon to the application programmer who is then relieved from worries about array sizes and limitations, because they can always be made larger when the need arises. It is there important to know whether selected hardware has an operating system that supports virtual memory.

Another distinction is between time sharing operating systems that can service multiple users simultaneously, and those that are single user systems. Some older

operating systems may be able to serve a few processes, one operating in a 'foreground' partition, and the others running in the background. Yet another category is one of 'real time' operating systems, specifically designed to monitor and control ongoing processes, such as in a factory or mill.

From a users points of view, the ease with which files are managed by the operating system is very important. A system that supports some type of hierarchical file directory system is preferred. The capability to organize files into a master directory with sub-directories and sub-directories within sub-directories can be extremely useful for organizing a complex application on a computer. Many physical and organizational systems in forestry can be decomposed into hierarchical systems which can then be reflected in the organization of corresponding files.

One operating system that is increasingly being used is the UNIX operating system. UNIX was first developed by Ken Thompson of Bell Laboratories. The first version was written in PDP-7 assembly language, but later versions have been written in 'C', a language developed by Dennis Ritchie of Bell Laboratories. With the decreasing cost of minicomputers the UNIX system has become very popular, and is currently one of the most widely available operating systems for a broad variety of machines. Because UNIX was not developed by a hardware vendor, it has been transported to a number of dissimilar computers, and so holds great promise for making application software independent of hardware based operating systems.

In the microcomputer field, there are a number of popular operating systems such as CP/M for 8 bit computers, developed by Digital Research and MS-DOS developed by Microsoft on assignment by IBM for its 16 bit personal computer. Some micro systems are capable of running several operating systems, in order to exploit the large collection of software available under each operating system. UNIX is also available on a number of micro systems.

As mentioned in the hardware section, many minicomputer vendors are expanding the bottom of their product line with a micro system. As a consequence, some operating systems usually only thought of in the mini environment, such as VAX VMS, now also have a micro implementation. The micro system may only have a subset of the functions of the full blown mini system.

There has been much development in higher level programming languages in the past years. Much application software in the field of forestry has been written in FORTRAN. This language is still widely used, and is one of the most portable languages around. Over the years it has progressed from FORTRAN II to FORTRAN 77 and beyond, an currently reflects comtemporary programming philosophy, through the inclusion of structured control statements. FORTRAN 77 also has a much improved character handling capability. Currently popular languages for scientific programming that may replace FORTRAN in the long run are 'C' and PASCAL. Each of these has features not found in FORTRAN that can be extremely useful. For instance Pascal has a provision for performing operations on sets, as well as a capability for handling pointer variables. Kay (1984) classifies languages rather arbitrarily by level. Each level supposedly provides more power for less effort. In his classification FORTRAN is a borderline lower level

language, while Algol and LISP are higher level languages. SMALLTALK and PROLOG fall in the very high level category, while VISICALC, EURISKO and a recent version of LISP are bordeline ultra high level cases. Pascal and 'C' probably belong to Kay's high level category.

In forestry, much use is also made of languages especially designed for simulating physical processes, such as SIMULA, GPSS, GASP IV, DYNAMO.

Much confusion exists in the graphics support area. The need for device independent graphics is obvious. However much work remains to be done before this will become an easy reality. Two standards currently compete for first place in this area. The CORE has been developed by SIGGRAPH, a special interest group of the Association for Computing Machinery in the U.S., while GKS (graphics kernel system) has its origins in Europe. The CORE has a three-dimensional capability, while GKS is mainly two-dimensional. It seems that GKS is currently more in demand than the CORE system. Graphics packages incorporating either standard can be purchased from various suppliers. Such a packages may have two ways for displaying data. The first is through a Virtual Device Interface (VDI), which translates graphics directly into terminal commands; the second is through a Virtual Device Metafile by which the graphics are translated to a device independent intermediate file, from which plots can be made on the device of one's choice.

Other basic utilities needed by application programmers are text-editors, document processors, debugging tools, format conversion software etc..

2.1.2.2 Database Systems

In this section we will present a review of database and database management systems (DBMS's) with a special emphasis on their applicability for forestry information processing. It is not meant to be a tutorial on database technology, but rather we like to give a brief overview of the current status of database systems, their characteristics, and potential use for forestry problems.

Many different sources for tutorial information on database systems exist. Some of the most well known are: Martin (1977), Date (1983), and a special issue of Computing Surveys, a publication of the Association of Computing Machinery, with articles by Sibley, Fry, Chamberlain, Taylor, and others (1976).

Fry and Sibley summarize the reasons for using a DBMS as follows:

1. to make available an integrated collection of data to a wide variety of users.

2. to provide for quality and integrity of the data.

3. to insure retention of privacy through security measures within the system.

4. to allow centralized control of the database, which is necessary for efficient data administration.

5. to make application software independent of the data.

For developing countries, the first reason is by far the most important consideration for the use of a DBMS. For example, in India, all reporting in the hierarchy of forest management is currently still accomplished with forms and ledgers. Some reporting steps may take several years, and hence there is a severe lack of feedback with regard to national and state planning decisions. The first step towards computerization would be to parallel the current system with a computerized equivalent. The full benefit of such a conversion would not be felt however until the data were made available for use through an integrated database management system.

Data independence is undoubtedly the second most important consideration for the use of a DBMS. Data independence implies that the programs which access the data can be independent of the way the data are physically or logically organized. This assertion holds to a degree, but in general it does free the programmer from a significant concern with the data structure. He no longer needs to be involved with the physical organization of the data, and logical changes in the data structure can be made without affecting his program. Unplanned situations and requests can be dealt with in significantly less time and with less effort. The use of DBMS systems is therefore also recommended where data are manipulated in many different ways for specific purposes such as may be the case with a forest inventory. The DBMS is then not used as a central data distribution system, but rather as a programming tool for an individual application. This type of beneficial use of a DBMS should not be overlooked.

Characteristics Of A Database Management System

As the name implies, a DBMS manages a database. Martin (1977) gives the following definition for a database:

'a collection of interrelated data stored together without harmful or unnecessary redundancy to serve multiple applications; the data are stored so that they are independent of programs that use the data; a common and controlled approach is used in adding new data, and in modifying and retrieving existing data within the database. The data is structured so as to provide a foundation for future applications development.'

Date (1983) simply defines a database system as a 'computer based record keeping system.'

Whatever one's definition of a DBMS may be, it has a set of shared characteristic features, namely:

1. recovery

2. integrity

3. concurrent use

4. security

5. data model

6. schema and data dictionary

7. data definition language

8. data manipulation language

9. report writer

In the following we will briefly describe each of these components, so that the reader will be able to judge whether a system is a true DBMS.

Recovery - Recovery refers to the capability to recover the database in case of failure. When the current state of the data is known to be incorrect, or at least suspect, the recovery procedure can restore the data to a correct state with a minimum of data or work loss. Many reasons for possible failure may exist, ranging from applications programming error to power failures. A recovery capability is essential, especially for developing countries, where power failures may occur frequently.

Integrity - Integrity is associated with accuracy, correctness and validity of the data. When cross-references exist, cross referenced data items must match, and be properly changed in case of an update. For instance, a species indentification may exists in two different records which are related to the same field plot. When either of these records is updated the species codes must still match, otherwise the integrity of the data is lost.

Concurrent Use - Concurrent use implies that a genuine DBMS can be simultaneously shared by several users. Not only must a transaction executed by a single user be processed such that integrity is maintained, but multiple concurrent transactions must not interfere with one another. To this end the DBMS must have a concurrency control mechanism. It regulates the sequencing and areas of access trough timing and locking devices such that the data integrity is maintained, and no updates are lost.

Security - Security implies the protection of the database against unauthorized disclosure, alteration or destruction. For instance, a system of passwords or keys maybe used to grant different kinds of access priviliges at different levels. Security measures are often not important and may even be bothersome when a DBMS is used for an individual, one user database application. However, for applications with multiple concurrent users security measures may be vital for the successful use of the system.

Considering the combined needs for recovery, security, integrity, and concurrent use, it is not an understatement to say that their adequate satisfaction may make or break a working system.

Data Model - A data model expresses how information can be represented and manipulated within the formal framework of a DBMS. Three broad categories of data

models are generally recognized: relational, hierarchical, and network. An overview of these approaches can be found in Date, Vol 1, (1983).

In a hypothetical database in which one stores data about forest stands, types of stand treatment, and applications of these treatments, one can have the following tables in a database based on the relational data model: STAND, with column labels stand number, species, origin date, location, and data values for each of these columns, one row per stand; TREATMENT with column labels: treatment type, treatment description, in each row describing the types of treatments that can be applied; and a table APPLICATIONS with application number, stand number, treatment type, treatment date (see Figure 1).

Each of these tables closely resembles a two-dimensional file or table called a relation, because when obeying certain constraints, they can be considered mathematical relations, as in mathematical relation theory. Other terminology from this field is used as well. Table rows are sometimes called 'tuples', and columns maybe referred to as attributes. The term domain is used at times to indicate the set of all possible attributes values for a given attribute.

All information in a relational database is presented in the form of two way tables, and the relational structure is therefore easy to understand and manipulate.

However a problem arises when considering attributes that may have more than one value. For instance, the species in STAND may actually be a composite of several species. In the relational data base this process is solved through 'normalization'. The stand record may be duplicated but with a different species value, and so on, until all species are accounted for. Data presented in this way are said to be in 'first normal form'. Other forms of normalization exist, all designed to promote the integrity of the data model.

In an hierarchical database, with a hierarchical data model records are organized as tree structures. For instance, in our example, the user may see as many tree structures as there are stands (one is shown in Figure 2). Each stand has a set of subordinated treatment records, and each of these, in turn, has a set of subordinated application records.

Hierarchical data structures are an obvious way to model the truly hierarchical structures of the real world. But frequently, relationships are not truly hierarchical. The model must then be enforced in an unnatural way, presenting unnecessary complications for the user.

Using the network data model, the data are represented by records and links between the records. Links may be used in hierarchical structures as well, but in a more restricted way. In a hierarchy, each set of records can have only one superior record (owner record), whereas in a network, there may be multiple superiors. Thus, the network approach allows for the modeling of a many-to-many correspondence that is not possible in a hierarchical model.

FIGURE 1

RELATION: STAND

Stand Number	Species	Origin Date	Location
22	Douglas Fir	1/6/1938	Block 2S
56	Hemlock	1/7/1922	Block 21W
37	Alder	1/6/1955	Block 7N
48	Douglas Fir	1/1/1980	Block 8W
.	.	.	.
.	.	.	.
.	.	.	.

RELATION: TREATMENT

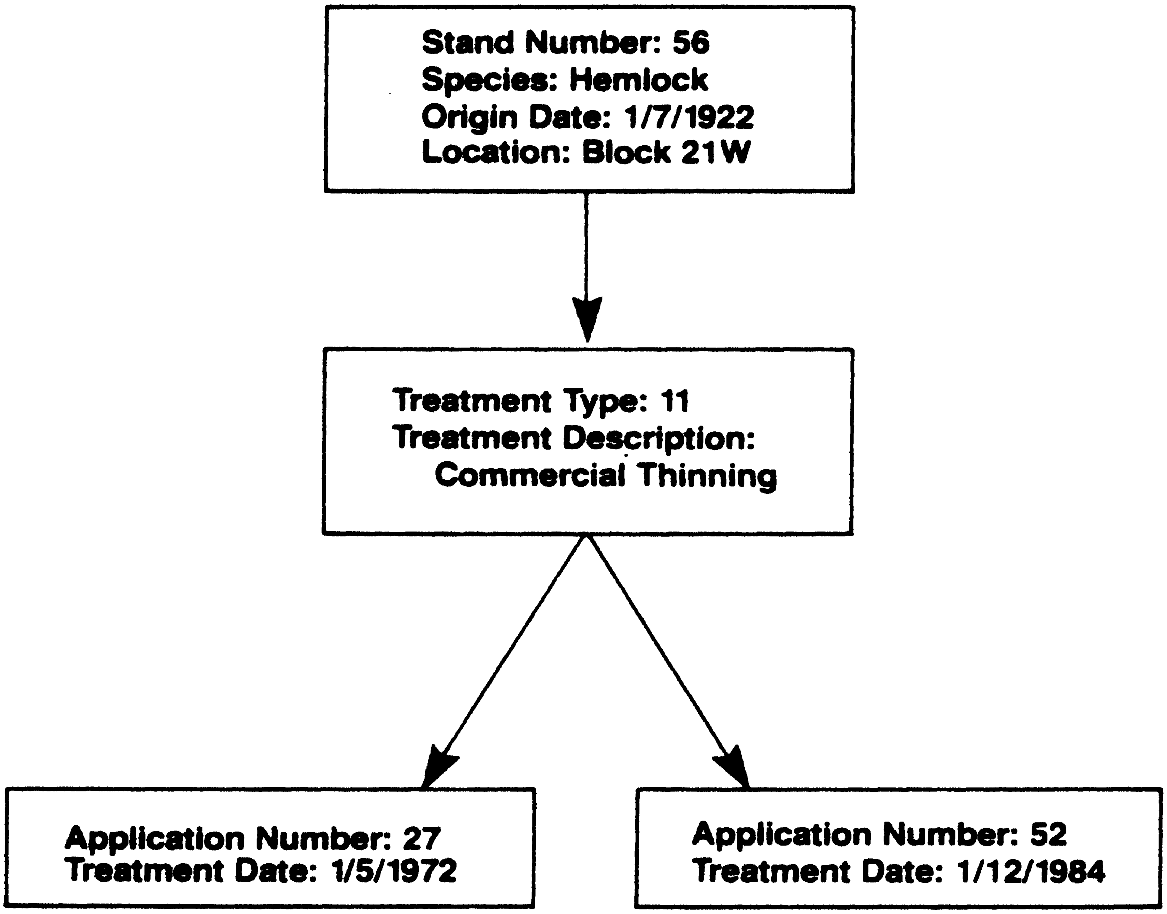
Treatment Type	Treatment Description
10	Pre-Commercial Thinning
11	Commercial Thinning
35	Fertilization
46	Pruning
86	Clearcut
.	.
.	.
.	.

RELATION: APPLICATIONS

Application Number	Stand Number	Treatment Type	Treatment Date
27	56	11	1/5/1972
28	37	35	1/7/1973
29	22	86	1/10/1984
52	56	11	1/12/1984
.	.	.	.
.	.	.	.
.	.	.	.

The Relational Data Model

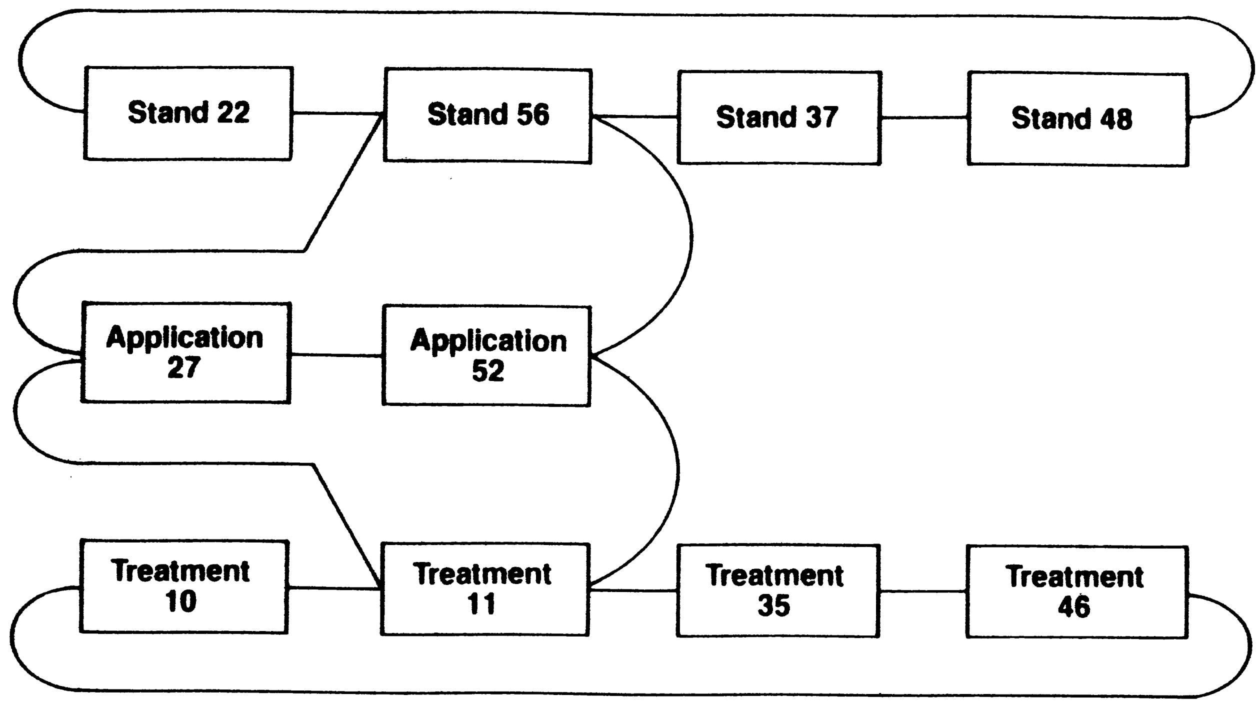
FIGURE 2



One Tree Structure for Each Stand

The Hierarchical Data Model

FIGURE 3



(not all treatments shown)

Each Node may contain other descriptive information

The Network Data Model

In a network model one can have records of the type 'connector'. In our example the connector occurrence can represent the association between a forest stand and a treatment, namely the treatment application. This connector can then contain the data describing the association, for instance the date of treatment application and the quantities (fertilizer) involved in the treatment. All connector occurrences for a forest stand can be placed on a 'chain', starting at and returning to the stand. Tracing this chain then provides a report of all applications. Similarly all connector occurrences for a given treatment type can be placed on a chain, starting at and returning to the treatment. The network organization for the example is illustrated in Figure 3.

A major disadvantage of the network approach is undue complexity. As Date (1983) points out: 'the source of complexity lies in the range of information bearing constructs supported in the network structure. The more constructs there are, the more operators are needed to handle them, hence the more complicated the Data Manipulation Language.'

Schema And Data Dictionary - Data stored in a DBMS must be described in a formal manner. There are essentially two aspects to the storage and organization of data: physical layout and logical representation. The DBMS user should not have to concern himself with the physical organization of the data, but the logical organization is most important.

The description of the overall logical database is referred to as a schema. Thus, the previously mentioned hierarchical, network, and relational models are all expressed in the form of a schema. They are often pictured in the form of a diagram, using blocks, with the associations between the blocks, such as the links in the hierarchical and network systems, presented by solid lines, and cross references drawn as dashed lines.

However, to initialize and update a database, one must be able to present the schema to the computer in text form. For this purpose the user or database administrator uses a data definition language (DDL) which has been specially designed to enter information about the attributes and their relationships.

Sometimes a distinction is made between an overall schema for the entire data base: 'the schema', and sub-schema's representing views of the data for individual users.

The data dictionary is an important tool for the database administrator. It can be a database in its own right, containing data about data. All the various schema's can be physically stored in both source and object form in the dictionary. The data dictionary will also contain cross- reference information.

Data Definition Language - With a real DBMS the user or the database administrator must have an easy way to define the attributes, variables, links, associations etc. The data definition language (DDL) serves this purpose. With the DDL therefore, the data model can be realized in a concrete way.

For a relational DBMS the DDL may simply consist of specifying the attribute names, their types and length (for text), as well as the relation names and the attributes that are assigned to the relation bearing that name.

Data Manipulation Language - Once a database exists, the application programmer or user must be able to make inquires from the DBMS through a set of instructions, which are most frequently cast in the form of a special language, the Data Manipulation Language. Sometimes this language is also referred to as the query or inquiry language.

Mostly, there are two ways in which such a language can be used: directly and indirectly. In the first way the programmer interacts directly with the DBMS in an interactive mode by typing in the DML statements and receiving answers from the DBMS. With the second method of use, the programmer can imbed DML statements directly in his application program. He may then have to first compile his program with a special pre-compiler, in order to translate the DML statements into a set of statements in the appropriate host language, such as FORTRAN or COBOL. In some cases a programming interface may be provided in the form of a set of DBMS routines that can be called directly from the application program. Such an interface does not qualify as a DML, however.

The DML or query language allows the user to interrogate the database to obtain answers to his problems, and is therefore the tool through which the objective for establishing the database and the DBMS is satisfied. For example in SYSTEM 2000, a hierarchical system of Intel/Commercial Systems Division, the user can interrogate the database through a query language, in which a query desiring all forest STANDS with treatments of COMMERCIAL THINNING and species of TEAK can be specified as follows:

> PRINT STANDNUMBER WHERE SPECIES EQ TEAK AND TREATMENTTYPE EQ COMMERCIAL THINNING

In some relational DBMS systems such as RIM, the same question could not be asked directly, but one would first have to perform a 'JOIN' between the STAND relation, the APPLICATION relation and the TREATMENT relation, to obtain a combined table with the required information which can then be accessed for the desired result with the following statement:

> SELECT STANDNUM FROM COMBINED WHERE SPECIES EQ 'TEAK' AND TREATTYP EQ 'COMMERCIAL THINNING'

The user must make perform the 'JOIN' between the tables because the relational system does not store any links or pointers as in the hierarchical or network data model. The JOIN produces the same result however by matching common attributes between tables. For instance, the stand number is both found in the STAND relation and the APPLICATION relation, and there forms a tie between the two tables.

The JOIN is called a relational operator. Other relational operators (among others) are SELECT and PROJECT. They are a part of a formal relational algebra, a system to manipulate relations, that can be imbedded in a DML as in the above example. Not all relational systems have this algebra as part of the DML, but instead use an approach referred to as a relational calculus. This is a more sophisticated but not necessarily more versatile system in which relational operations are performed automatically. For example, the Relational Technology Inc. INGRES system uses the relational calculus.

Report Writer - A versatile, easy to use report writer is a veryimportant component of a DBMS. Many programmers hours may be invested in the physical appearance and layout of a report using basic progamming languages such as FORTRAN or COBOL. With a flexible report writer reports may be produced in a fraction of time used otherwise. This is not to say that all reports can be produced with a report writer, one may still have to resort to a basic language for specialized reports, but substantial economies can probably be realized in the overall effort. The possible use of a report writer may be a consideration for using a DBMS in a single application such as the data processing effort for a forest inventory.

For a relational DBMS, in its most simple form, the report writer may simply present the data stored in a relation, with appropriate column headers. In a more sophisticated version the user may be able to specify how the data will be organized on the page. He may be able to define page breaks, totals, subtotals, value labels, page headings, expression values, all with easy-to-use layout directives.

The report writer may also be a means for summarizing data from the database for transfer to another system, such as a statistical or another applications package.

Current Trends

Of the three types of data models described, the relational model is becoming more and more established as the data model of the future. A number of relational DBMS systems is now commercially available, both on larger minicomputers and on micro-systems. This trend is due to the appealing qualities of the relational model, namely simplicity, because of the two dimensional tables and the absence of pointers and links, and the relational operators or relational calculus which can be employed in a convenient data manipulation or query language. Together they provide a flexibility that is not found in the other data models. With the relational and network models the user is forced to anticipate most uses for the database, and provide the appropriate structure accordingly; if a type of use is not thought of in advance, the DBMS will not be able to cope with the requests for that application. With the relational data model, there is far more leeway for improvisation to deal with ad hoc inquiries. This does not mean that the relational model should not be thought out carefully before establishing the database however: a careless implementation can lead to data inconsistencies, especially when updates are applied.

One traditional objection for the use of relational systems has been a lack of speed: because of the absence of pointers, relations must often be established at the time of inquiry. But relational DBMS systems are becoming faster as there evolution progresses. For instance, Relational Technology, current distributor of INGRES, claims a 10 times speed advantage over the university version of INGRES, first developed at the University of California at Berkeley. Relational DBMS systems are currently also being implemented in a hardware, in the form of a back-end database machine, an example is the Britton Lee Intelligent Database Machine (IDM). It isolates much of the data processing from the host computer, thereby providing very fast response times. This technology is currently reliable and becoming more widely accepted.

Systems On Mini And Larger Computers

Three representative examples of of database systems using the three data models described previously are given in Appendix B. There are currently a great many systems on the market (93 are listed in a recent survey), and there is an especially explosive growth in the microcomputer field. The systems described are therefore only examples: there may be others that are functionally equivalent or superior. Systems decribed are: System 2000, SEED, and RIM.

Systems On Microcomputers

Some people are of the opinion that microcomputers systems have not yet fully matured into full-blown database management systems, and that much current use is 'hobby type use'. Nevertheless, there are probably many meaningful functions that be performed with a micro-system. In Appendix B, the following systems are described: dBASE II, MicroRIM, Knowledgeman.

A recent survey of micro databasemanagement system was compiled by Bond (1984). Bond comments, that whereas a few years ago the only powerful systems were dBASE II, Condor, and a few others, today the choice is so wide as to be bewildering. He compares 19 file management programs, 22 relational database management programs, and 6 multiuser DBMS systems.

Forestry Application Examples

As mentioned earlier, DBMS systems can be extremely beneficial when a large pool of data can be simultaneously shared by a number of users, but can also provide an advantage for a single user because of its general data handling capabilities, such as report writing. For some types of forest inventory, such as a small one-time inventory, there may be but a single user who still could benefit by using a DBMS suitable for the task, because of its time saving features. For larger inventories with multiple users of the inventory data, such as may be the case for planning inventories, the use of a DBMS can give a very pronounced advantage. The DBMS then provides integrated data structures, separate access languages for programmers and non-programmers, and data integrity and security functions. In the U.S. Moser (1976) demonstrated the application of a DBMS for operational maintenance of compartment inventory records. Murphy (1981) designed two trial databases with data from the state of Minnesota's inventory for management of the Aspen-Birch unit. In his report 'Database Management: a Forest Inventory Application ', he discusses the methods used and results obtained with System 2000. In designing the database he stresses the need for anticipation of the types of reports, and for estimating its total physical size. There exists a trade-off between processing flexibility and storage costs. In this respect, a hierarchical system is far more rigid then a relational system. Once the database has been defined and the initial data have been loaded, it is essential to have adequate user documentation in the form of a user's guide and a database manager's guide.

Murphy created a plot summary database and a tree detail database. For the plot

summary database he selected a simple two-level hierarchy. The first level contains plot attributes such as acres, stand-age, forest type, site index. The second level contains species group attributes such as species group code, number of trees per acre, and growing stock volume. The relation between a plot and its species groups is one-to-many: a characteristic of the hierarchical system. Two types of requests could be obtained with this database: simple retrievals of data, and more complex reports. An example of one query was to obtain a report of the range of site index values for each of 14 forest types. In another example it was used to satisfy a request for a formal report consisting of acreage summaries for forest types by site index, age and basal area categories. With the addition of growing stock volume to these data, a set of inputs for a linear programming module was easily obtained.

The tree detail database was also designed around a two-level structure. The first level consisted of the same data as for the plot summary database. The second level was made up of individual tree attributes rather than species group attributes. The relationship between the levels was also one-to-many. The same formal summary reports could be produced as from the plot summary database, but costs could be up to 10 times higher. The use of a special plot summary database was therefore well justified. The tree detail database is suited for information requests requiring individual tree data, such as required for tree growth projection systems. Murphy investigated some alternative ways of storing and processing the tree data. One was to use a packed binary sequential file, a second was to create an intermediate file from the tree detail database, and a third was to use the plot summary database to access the sequential tree file. The third alternative proved to be most economical.

For plot summaries, he concluded that the System 2000 DBMS had advantages over the processing of sequential files. Savings in personnel time paid for establishing the database. The software provided previously unavailable reporting capabilities and user access to the data. He did not think the tree detail database to be appropriate for unit wide summary reports however. The potential of such a database lies in research applications, for which the cost-effectiveness may be hard to determine. He concluded that the increased accessibility of the survey data would probably justify the cost of the tree-detail database for work units with research assignments.

System 2000 is currently widely used within the United States Forest Service. Since 1975 it has been used for storing and analyzing timber sale accounts. Most national forests have System 2000 databases for timber management. Other database exist for recreation potential, soil types and wildlife data. One or more stand or tree growth models routinely project, with inputs from this system, implications of alternative silvicultural treatments. National Forests also use the system to provide mutiple objective optimization programs (FORPLAN) with inputs.

One interesting application has been the incorporation of the system into an Integrated Pest Impact Assessment System, together with a Geographic Information System (MOSS) and a forest and socioeconomic prediction modeling program (Daniel, White, and Hunter,1983). Here System 2000 was used to augment the limited capacity of MOSS for storing timber stand attribute data, while MOSS provided the map based component of the system. The prediction and socioeconomic programs consisted of a

stand growth model, a pest model for projecting growth reductions and mortality due a particular pest, and socioeconomic equations relating forest characteristics to economic recreation and aesthetic values. With this system it was then possible to consider the impact of different pest management plans on a stand by stand basis.

Garret, Rogers and Prosser of the United States Forest Service (1981) describe a program for developing multiresource management guidelines based on multiresource inventories and database management, in conjunction with a multiresource management planning model. In using System 2000 they discovered that it does not fully meet DBMS needs for interactive muiltiresource models. Its greatest weakness is its inability to interrelate the various components of the hierarchical trees of the database. They therefore judge System 2000 as limited for a research DBMS. They conclude that because of its widespread use, it should be considered however, and that the inadequacies can be overcome by writing peripheral software.

In a field office of the Southeastern Forest and Range Experiment Station in Starkville, Mississippi, the Unites States Forest Service, is using a relational database system for the management of inventory data. The system used is a VAX 11/780, and the database management system is INGRES of Relational Technology Inc.. Large data bases have been implemented, ranging in size from 25 to 50 Megabytes. The largest relation contains all the tree data for a state: such a file may have over 90,000 rows with 102 variables per row. The field office maintains data for six states. They are now in a process of discovering how to manipulate these large databases, and currently have some problems extracting the data from the database for processing by statistical packages such as SAS.

2.1.2.3 Statistical Systems

Statistical software packages were first developed in the 1960's. As with DBMS systems, they resulted from the realization that it is more economical to have dedicated systems for frequently occurring tasks. Today however, one of the three main approaches for solving statistical problems is still the use of a basic language such as FORTRAN. In most cases a better use of human resources can be made by either one of two other approaches, namely the use of a 'mainframe' statistical package or a more interactive language such as APL.

Two types of statistical analysis are needed in forestry information processing. The first is to handle to the data for a forest inventory. The second is to deal with the requirements of other miscellaneous tasks for forestry disciplines such as ecology, wildlife, soils, growth and yield, forest mensuration etc. Forest inventory and its associated statistical analysis is a very specialized field based on sampling survey theory. Its needs cannot be met by statistical packages such as BMDP, SPSS, etc. Some general sampling survey packages that can be applied in this area are mentioned in the section of this report entitled ' Inventory Data Processing.' Many specialized statistical procedures exist in the other disciplines as well, but there is a much better opportunity for using packaged software. The use of standard functions for solving problems with analysis of variance or multiple regression can be especially beneficial. Not only can the systems software be used to get the desired results, but the system documentation and

the way in which the data analysis is presented can be valuable for statistical training.

Systems On Mini And Larger Computers

The are currently a number of 'mainframe' statistical packages on the market. Best known are: Minitab, SPSS, SAS, and BMDP. Originally developed for larger 'mainframe' computers, these packages are now also available on minicomputers. They have been closely modelled on the idea of sequential processing of a series of records on punched cards or magnetic tape. Relatively recent user guides to BMDP and SAS still picture the user input as a card deck. A recent system developed by American Telephone and Telegraph Bell Laboratories named 'S' is not burdened with this inheritance of the past, and allows highly iterative and freewheeling data analysis, as dictated by the data. S is available under the UNIX operating system.

The current trend for the larger systems is to provide more periheral facilities for the user in terms of graphic output and file handling. The card deck oriented command sequences are giving way to more sophisticated command languages. SAS especially is known for its command language that allows for such features as data editing, subsetting of datasets, concatenation, merging and file updating, as well as handling of multiple record types with different field compositions. An attempt has been made to develop a pre-processor for translating SAS statements into the BASIC language for use on a microcomputer (Bass,1984). At FAO Headquarters, Singh and Lanly (not dated) are proposing to use the SAS package for updating, retrieval and reporting in the context of a global forest resources information system. They give the following reasons: it is a simple user oriented package that can be easily learned by foresters and other investigators with little knowledge of computer programming; it can be operated in both batch and on-line modes; it is currently used extensively within the Forestry Department and by other technical departments of FAO. For disavantages they cite that: it has a limited capability as a inquiry language; it is not available in many areas of the world. However they state that their choice of SAS appears to be a good choice within a short time horizon of five years. While the FAO system is not a spatially oriented system as such, the use of SAS has also been considered for this purpose by Flint and Spear, who built a highway control system in the U.S. State of New Mexico, based on SAS and the graphics package DISSPLA by IISCO.

Appendix B has system descriptions for Minitab, SPSS, BMDP, and SAS.

Systems On Microcomputers

For statistical computing on microcomputers there is currently a wide choice of available packages. At least one of the mainframe systems (BMDP) is available on a microprocessor (StatCat). Carpenter, Deloria, and Morganstein, recently conducted an in-depth review of 24 statistical packages for microcomputers. Their categories for comparisons were the following: operating systems, hardware requirements and price; general program features; data limitations (case handling, missing values, significant digits, numeric range etc.) documentation features, data management (keyboard access, foreign data access, file documentation, edit and print features); data processing (transforms, sorts case selection, file joining); summary statistics (means, moments,

coefficients, order statistics); graphics (histograms, scatter plots, pie charts, etc.), nonparametrics and tables (Kolmogorov-Smirnov, Mann-Whitney, Wilcoxon, etc.); linear models (regression, t-tests, ANOVA, etc.); accuracy of regression coefficients; time series and ratings of main features. Some of Carpenter's, Deloria's and Morganstein's observations based on the result of comparisons in these categories are:

1. The best of the micro-packages are eminently practical for professional serious applications.

2. The best of the packages equal or exceed mainframe packages in accuracy.

3. For small data sets analysis on a micro is definitely preferable.

4. Micro-packages have much better ties with other applications packages.

5. Users may need to buy more than one package to get all needed features.

6. Documentation of large data set capability is poor, some packages handle multidisk data sets, consultation with dealer is advised.

7. Packages are primarily written in Basic and run slowly.

8. They are easier to learn and use than mainframe packages.

9. All packages have some annoying features, the most serious problem is inadequate error handling.

10. No clear favorite emerged form the comparison.

The final opinion reached by Carpenter, Deloria and Morganstein is that many packages merit serious consideration, and that the users choice should be based on his preference for certain features.

In Appendix B, ABSTAT, is described, because of its compatibility with the popular DBMS dBase II software.

2.1.2.4 Geographic Information Systems

Basic Concepts

In forestry, geographic information systems are assuming a more and more important role. Quite often they are referred to as a 'GIS'. As the name implies they provide information about the geography of an area. However this holds true for an aerial photograph or a Landsat image also, but these are not commonly thought of as a GIS. Wherin lies the difference ? Obviously, a GIS is a computerized system. However, an image analysis system in which aerial photographs or Landsat images are stored satisfies this criterion, but need not be a GIS.

The real difference lies in the fact that the GIS has been designed to optimally satisfy recurring geographic information needs of the user. This means that the system has been optimized to store, retrieve and update geographic information concerning the users area, and that the system has been programmed to optimally process this information on demand, in a format most suitable to the users application. In fact, the system should contain an accurate and up-to-date model of the users area.

With an image analysis system one can make a Landsat classification map that may be of interest to an agency or a user community or a particular individual. With a GIS on the other hand, one can integrate various types of geographic data and maps to answer questions such as: what treatments where applied to a forest compartment during the past ten years, or where does a certain covertype soils combination occur, or one can make a map of all ownerships of a certain type in the area.

Recently, for many original image analysis systems, 'GIS' functions have been added to the software. These area functions that alter and combine image in a geographic sense, to provide answers and maps required by the user of a GIS. An example is the capability to generate a buffer zone around lineal features for the purpose of evaluating their impact on other features. Such functions still may not qualify a system as a GIS, because they do not necessarily optimize the system for the continous storage, retrieval and updating of spatial information for a specific purpose. A system of this type is more of an analysis system than an information system, therefore, we would like to introduce the term Geographic Analysis System as opposed to Geographic Information System. We will then distinguish three categories: image analysis system (IAS), Geographic Analysis System (GAS), and Geographic Information System (GIS). These distinctions are not found in other literature, but overall, they will be helpful in discerning the true nature of a system, when too many systems are grouped under the GIS umbrella.

The main concept of a GIS is that one maintains a set of spatially registered data layers, maps or overlays. This concept is illustrated in Figure 4. These layers can be stored in either raster or vector (line) form.

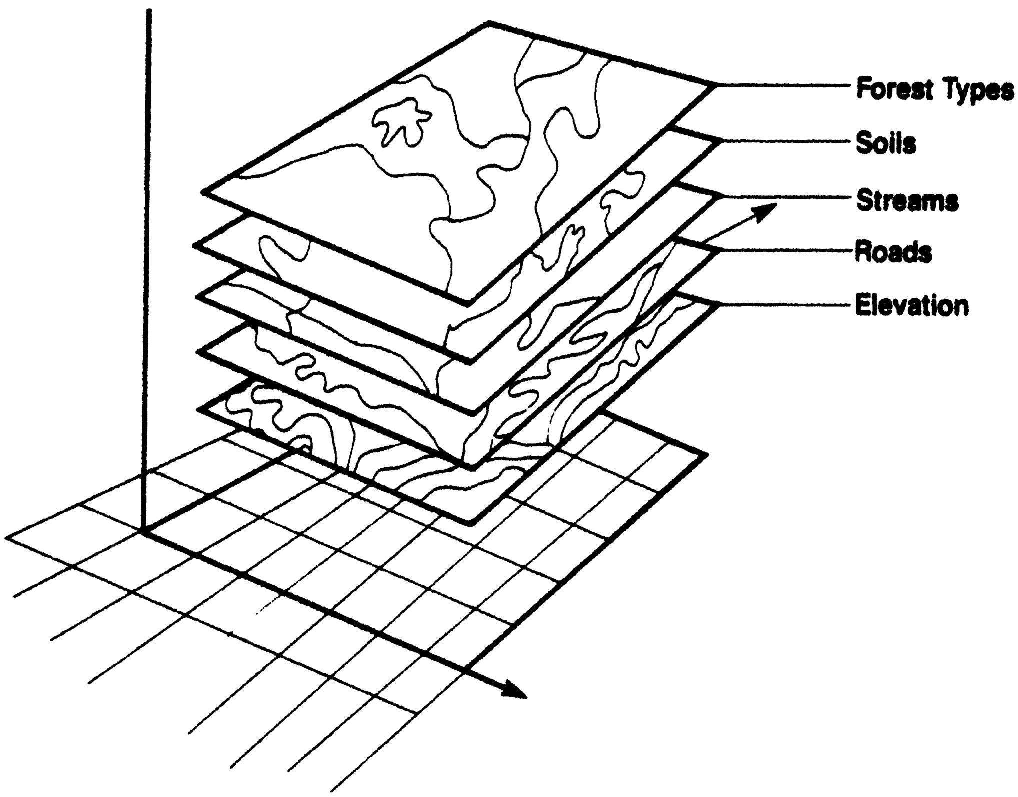
In a raster or cell based system, the map is represented by a rectangular array of rectangular or square cells, each with an assigned value. In a vector based system, the line work is represented by a set of connected points: the line segment between two such points can be considered a vector (not to be confused with a vector of bands in an image analysis system). The coordinates of the points are explicitly stored, the connectedness is implied through the organization of the points in the database.

The characteristics of the two system types are easily grasped by considering advantages and disadvantages of their data representations.

The advantages of a raster based system are:

o The geographic location of each cell is implied by its position in the cell matrix (image). The matrix can be stored in a corresponding array in the computer provided enough storage is available. Each cell can therefore quickly and easily be addressed in the machine according to its geographic

FIGURE 4



A GIS conceptualized as a set of geographically registered data layers.

location.

- o Since the geographic location is implied in the cells position, the geographic coordinates of the cells need not be stored.

- o Neighbouring locations are represented by neighbouring cells, therefore, neighbourhood relationships can be conveniently analyzed.

- o A cell system accomodates discrete data (such as forest types) equally well as continuous data (such as digital terrain models), and facilitates the intermixing of the two data types.

- o Processing algorithms are much simpler and easier to write than for vector based systems.

- o Map unit boundaries are inherently presented by different cell values ; when the values change, the implied boundaries change.

- o Raster based systems are more compatible with raster based output devices, such as line printers and many graphics terminals.

- o Raster based systems are compatible with systematic type inventory procedures.

The disadvantages of raster based systems are:

- o Storage requirements are much larger than those for vector based systems.

- o The cell size determines the resolution at which the resource is represented. It is especially difficult to adequately represent linear features.

- o Most often, image access is sequential. This means that one may have to process an entire map just to change a single cell.

- o Processing of associated descriptive data is more cumbersome than with a vector based system.

- o Input data are mostly digitized in vector form. One must therefore execute a vector to raster transformation to convert digitized data into a form appropriate for storage.

- o It is rather difficult to construct output maps from raster data.

Advantages of vector based systems are:

- o Much less storage is required than for raster based systems.

- o The original map can be represented at its original resolution.

- o Resource features such as forest types, roads, streams, inventory plots can be individually retrieved and processed.

- o It is easier to associate a variety of descriptive resource data with a single resource feature.

- o Digitized maps need not be converted to raster form.

- o Stored data can be processed into line-type maps without a raster to vector conversion.

Disadvantages of vector based systems are:

- o Locations of the vertex points need to be stored explicitly.

- o The relationship of these points must be formalized in a so-called topological structure, that may be difficult to understand and manipulate.

- o Algorithms for accomplishing functions that are the equivalent of those implemented in raster based systems are far more complex, and implementations may be unreliable.

- o Continuously varying spatial data cannot be represented as vectors: a conversion to raster is required to process data of this type.

Both raster and vector based systems can be purchased as 'turnkey' systems. The term 'turnkey' system refers to a system consisting of a set of integrated hardware and software that is ready to perform its intended application without further user modification, or any user knowledge of the inner workings of the system. The term originates from the pre-computer era when it was used to describe a contractor developed building or project for sale when completely ready for occupancy or operation. For computerized systems today, it emphasizes the total integration of hardware and software for a specific purpose.

As time goes by, the term turnkey is becoming less and less applicable to describe a large number of systems that are on the market. They are becoming less rigorously structured, with more and more component parts that can be replaced by other parts; while the software is becoming more and more transportable between different types of computers. This is a healthy trend, because a system can be better adapted to the users needs, but it also leads to more confusion for the potential buyer.

In the remainder of this chapter, we will discuss both raster and vector based systems. The last section of the chapter is devoted to current trends. We will organize the material according to eight major systems aspects, which according to our experience are the points of interest for a variety of systems:

- o data organization

- o database functions
- o input
- o query and analysis
- o display
- o reporting
- o user interface
- o hardware

Data organization addresses to the vector versus raster issue; database functions covers topics such as use of operating system or real data base system; input covers digitizing and input from other sources; overlay is a query and analysis topic; user interface discusses menus versus command mode and other methods of control; display touches on graphics output, while reporting is concerned with tabular reports. Hardware is the last category: to any system observer it is the most prominent, but for the experienced user it is a part of his environment.

Raster Based Systems

In this section the major system aspects of raster based systems will be described. A set of typical software functions is presented in Appendix A, where they are also applied to an example, that will hopefully provide some insight into the types of problems that may be solved with raster based systems in developing countries.

Data Organization - The most important aspect of raster based systems is the selection of the cell as the basic data element. The cells are stored as an ordered set of numbers. They are organized in a uniformly spaced pattern of perpendicular rows and columns, that can be superimposed over a geographic area. Each cell can then be looked at as a rectangular (or square) parcel of land, and a map value can be associated with the cell to characterize that geographic location. Cells may also be thought of as sample points located at the center of the parcel (systematic forest inventories).

The real advantage of cell systems is of course the fact that the position of the cell record in the file is a direct function of its geographic coordinates. Retrieval can therefore be very rapid, although more often than not, systems are not organized to take advantage of this. Usually one must process an entire image or map to access one particular spot of interest.

Determination of a cells address (place of storage) directly from the geographic coordinates is called hash coding. An example of such a function is:

$$N = (YC - 1)(XMAX) + XC$$

where XC and YC are the coordinates (in raster increments), XMAX is the number of cells in a row of the grid, and N represents the sequential record number for the cell in the file. For example, N = 28 when XC = 4, YC = 5, and XMAX = 6 (see Figure 5).

This function demonstrates an arrangement of cells, whereby they are physically stored in a single column, and the cells address is computed from the logical row and column location.

Most frequently, the two-dimensional nature of the grid will be reflected in the storage method however, with the map or image stored in a two- dimensional array, or the rows being represented by records in a diskfile.

Two alternative storage methods for cells exist. They prohibit direct cell access, but are often used in sequential processing. The first is called run-length encoding. Its aim is to reduce storage requirements by exploiting the correlation between adjacent cells in a row. Rather than storing a run of say 100 cell values that are identical, one stores the count (100) and the value. Many variations on this basic theme exist.

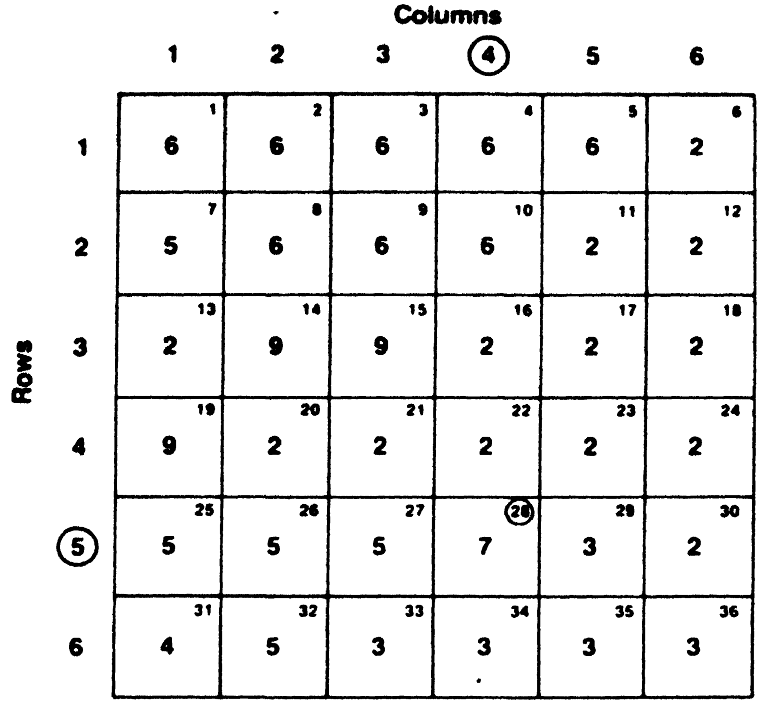
With the other storage method, the cell column and row number are explicitly stored, together with multiple values or attributes for the cell. Only cells where these properties change are stored. One advantage of this method is that only cells of interest are stored; no space is wasted to fill a rectangular area, when the study area is irregular.

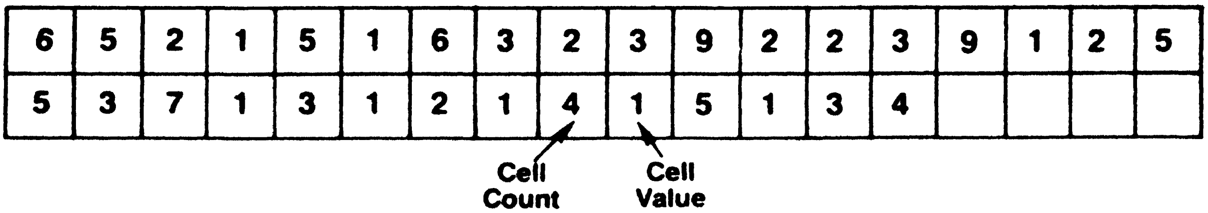
A cell is just one element of the grid. In image processing systems this grid is usually referred to as an image. For systems that are more GIS oriented the name map or overlay may be used. Other terms that are sometimes used are : dataset (a rather vague and confusing term); layer (one grid may be part of a stack of grids); theme (each grid in the stack may have a different thematic content). In this report we will use the term layer when we refer to the different layers in the stack. However, when a particular rectangular area in such as layer is referred we will call it a map or an image. Our usage of the word map may be slightly confusing because the traditional map usually has more than one layer: however, the simple word map is preferred over other more artificial terms such as geoblock, control unit, tile, etc.

A registered set of maps also goes by various designations. In the image processing tradition one speaks of a multi-band image. Such an image may be organized in different ways. All bands for a given cell may be stored together, as a vector, pixel stack, or slice (band interleaved format) or the bands may be stored completely separated (band sequential format). A disadvantage of the former method of organization is that it may be difficult to add more layers than were originally allocated, whereas with the latter method bands can be freely added. Since this occurs frequently during analysis of the data, the latter method is preferred for GIS systems. Another name for a registered set of layers is 'database', or one may irreverently call it a sandwich. The difference between the band interleaved and band sequential formats is shown in Figure 6.

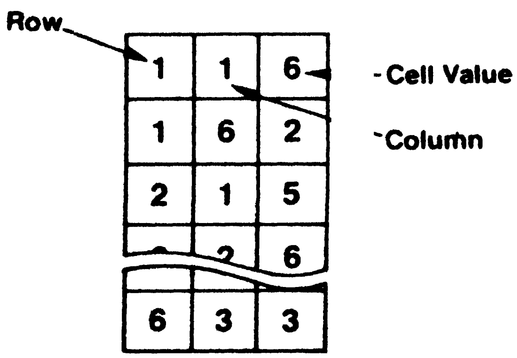
Grid layers may be of different types. Two or three are recognized. The basic distinction is between discrete and continuous layers. Polygons that represent areas

FIGURE 5





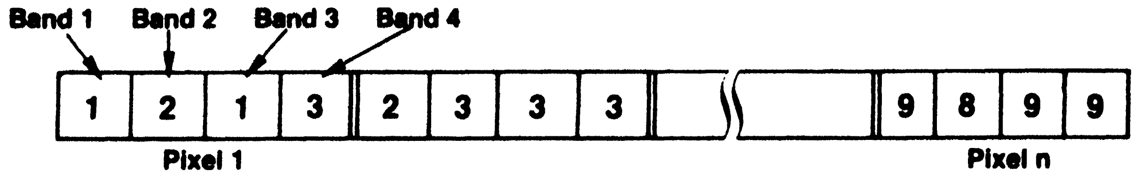
Run Length Encoding



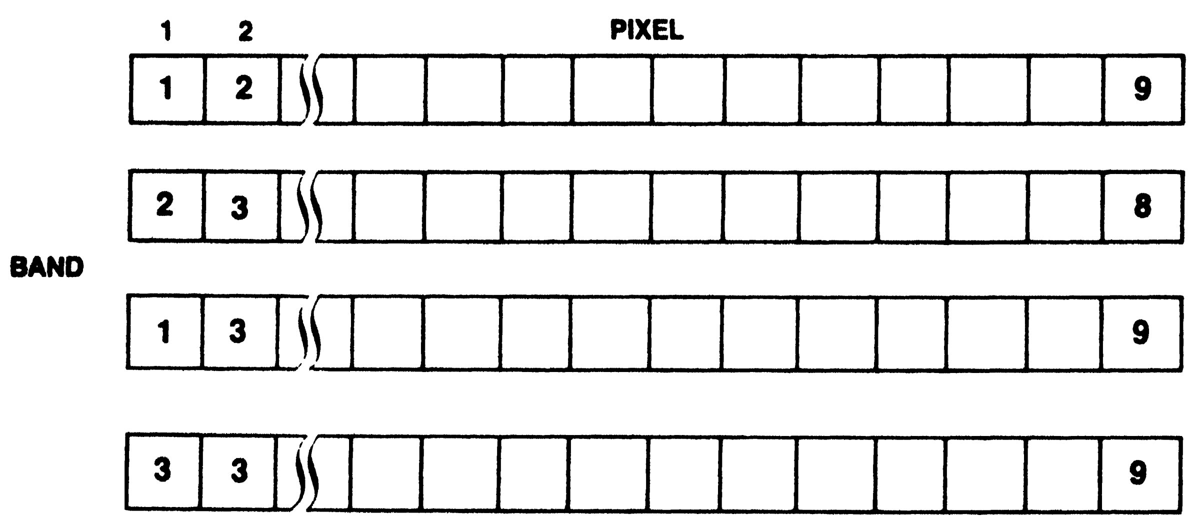
Explicit Row Column Coding

Three Raster Data Organizations

FIGURE 6



Band Interleaved



Band Sequential

Two Organization of Multiple Images (Bands) for Same Area

with homogeneous characteristics are processed into discrete layers in which there are extensive groups of pixels with the same value, representing the same type. Continuously changing characteristics such as elevations are represented by continuous layers. In these layers grid cell values are likely to change from cell to cell, with some statistical predictability.

Two basic data types exist in a GAS/GIS: spatial data and attribute data. Spatial data contain the positions of features in the resources, while attribute data represent other physical facts about them. For instance, soil type, soil depth, and soil texture may all be attribute data of a soils polygon for which the spatial data is a list of point coordinates of the polygon. One can easily see how one might store one record with coordinate data followed by another record with attribute data.

In a raster based system, more complex attribute data are not as easily accomodated as in the above example. Single value attribute data can of course be the cell values. There are two drawbacks however. First, the cell value must be numeric, and so non-numeric attributes such as alphanumeric type codes must be translated into unique numbers. Second, this number must fit within the allotted range. This range depends on the amount of storage space set aside for each cell. With one byte per cell, one can only use 256 different integer values. For two bytes this number increases to 65536. When four bytes are allocated. over 4 billion integer values can be used, or one can use 'real' values with a floating decimal point. However, when more than one attribute needs to be associated with each cell, the user is often left in the lurch. Most system functions may only operate on one cell value. A common solution to the problem is to store a pointer as the cell value, pointing to a row in a attribute table, in which the applicable attributes for the cell are found. However, the norm is that the system does not automatically handle this link and the associated data. Multiple attribute data may even be stored in a different data base, or be associated with a statistical package.

One system does accomodate multiple attributes with a so-called multi-variable file format. Row and column numbers of the cell are stored with a list of variables for the cell. Cells are only stored when at least one of the variables is different from the previous cell. This file may then be used in the same way as a stack of single variable grid layers.

The three different raster data organizations are illustrated in Figure 5.

Database Functions - Earlier, we defined a GIS system as a system that has been programmed to optimally process information on demand, and present it in a format that is most suitable for the users request. This definition certainly implies that the information is managed by a special database system, that can quickly retrieve, store and delete areas of interest. Two kinds of information exist in a raster based system, the grid layers (maps) and the associated attribute data. Attribute data can be stored in a conventional database management system, and this is currently a trend in the development of GIS systems. However, spatial data are not readily accomodated by a conventional DBMS because of the two-dimensional nature of the data. This is especially true for raster data. For this type of system, the most the user can expect is a map cataloging service. It will automatically keep track of the various maps that are stored in different system files, so that the user only has to remember the map name. It may also

indicate whether a map is on line, or has been archived, whether it is a master map, or a workmap, etc. However, gridlayers are not automatically registered, nor are there database functions that will automatically mosaic maps, or retrieve arbitrary areas of interest.

A second important function of the DBMS is security. Some raster systems, although they do not operate with a DBMS, do offer security measures in the cataloging service. For instance, certain maps may be protected, so that they cannot be changed, or they may be 'exposed', allowing the user to change them at will.

An important function of the cataloging service is to archive and de- archive maps. Each gridded map may occupy much on-line storage, and therefore maps may have to off-loaded to tape storage, and be restored from tape storage. The cataloging service should know the status of each map at all times.

Input - Raster data either originate from a data acquisition device that produces raster formatted data, or the data must somehow be converted to raster by the system user. Landsat data is probably the most well known source of raster data, but there are other satellite based remote sensing devices such as the AVHRR (Advanced Very High Resolution Radiometer), the Thematic Mapper, and in the near future Spot.

Already existing map and photographic products can be rasterized by scanning the product. A variety of scanning devices exists, such as drum scanners, flying spot scanners, microdensitometers, charge couple devices, and video cameras. However, none are very regularly used to input already existing line type maps. Maps must be specially prepared, and the scanned work may require much editing to become 'clean'. Also, devices that do not significantly degrade map accuracy are extremely expensive. Because of the investment in equipment, much of this type of work is performed on a service basis in many countries.

Most of the input of line map data is accomplished by digitizing the maps, capturing the line data in vector format. The vector data is then converted to raster form with a vector-to-raster software conversion package. Such a program typically scans the vector data internally in the computer, while detecting the crossings of the scan lines with the digitized vectors. Cell values are then generated based on these crossings and line attributes encountered.

Before this type of software became widely used, many agencies and users resorted to hand encoding methods, in which acetate grids were overlaid on the maps, and values of each cell, or 'run of cells' were encoded by hand on input forms, which were then keypunched. Computer assisted methods for doing this work were also developed: for instance, one might key the data directly into the computer, while interpreting the cell values on the map.

Whatever technique is used, input is persistently one of the most problematic areas of any GIS/GAS system. This is particularly so for raster based systems. Input is further complicated when considering multiple grid layers, because they must be properly registered. This problem does not exist when all layers are derived from the same basic

map, but many times different data sources are involved. When these data are all line data from maps, quite often the digitizing system can be used to register the maps to one base map or to a common coordinate system, before the vector to raster conversion. Performing the registration with different scales and projections in the raster domain is probably the most time consuming and complicated way of achieving registration. The input data sources may then have to be transformed and resampled, and this requires considerable amounts of computer resources.

Analysis - Once a set of registered gridlayers have been createdand found to be in good order, analysis can take place. This phase is also referred to as query or inquiry, but these terms are more often used in vector based GIS systems, and so we will use the term analysis. Most often, the analysis capabilities of a system are described in terms of available 'functions' or 'commands'.

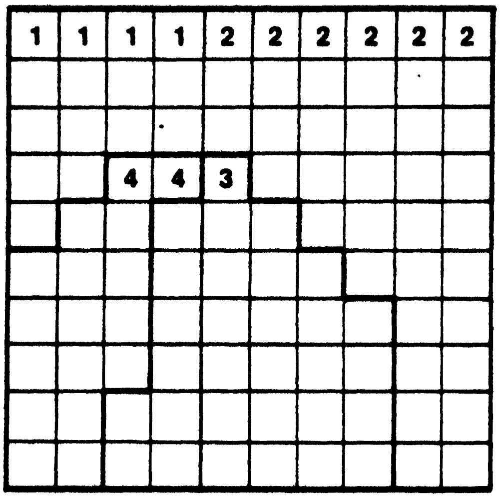
Each command usually invokes an independent program or system module. Some systems may offer the user the choice of over 70 different functions. One must always keep in mind however, that the number of functions is not at all indicative of the quality of the system. For example many functions may be bookkeeping or database functions that detract from the intended use of the system. This issue will be further addressed in the section on user interfaces.

Raster system functions may be divided into two broad classes: 1) system and utility functions, and 2) analysis functions. The first class can again be divided into three groups: data storage (database), input/output and program control. The analysis functions are sometimes divided into four categories: 1) reclassification, 2)overlay, 3) distance functions, 4) neighbourhood functions. In the following we will briefly explain the nature of the functions in each class. Example functions are given in Appendix B; they are based on those found in the MAPS system, part of the US Fish and Wildlife Service AMS/MOSS/MAPS/COS system (see the section on systems descriptions).

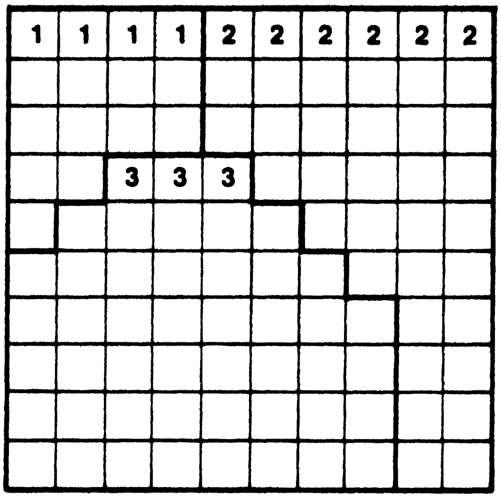
- o Reclassification Functions. The purpose of reclassification functions is to reassign or reclassify values of an existing map. As a result the boundaries inherently represented by the different gridcell values will change. An example of a reclassification function is shown in Figure 7.

- o Overlay Functions. One difference between reclassification and overlay functions is the number of layers. Reclassification functions operate on one input layer, overlay functions operate on more than one input layer. The input layers are 'overlaid', and a new grid layer is created of which the cell values are a function of the corresponding cell values of the input maps. Depending on the specified operation, the function may be a logical, arithmetical or statistical operation, or be a combination thereof. The overlay functions always operate on a cell by cell basis: for each cell the input values are transformed to one output value, usually with a simple arithmetic operation such as addition, or a logical operation such as 'or'. This basic operation is repeated for all cells in the map. An example of an overlay function (similar to the MAPS CROSS function, see Appendix A) is shown in Figure 8.

FIGURE 7



Input

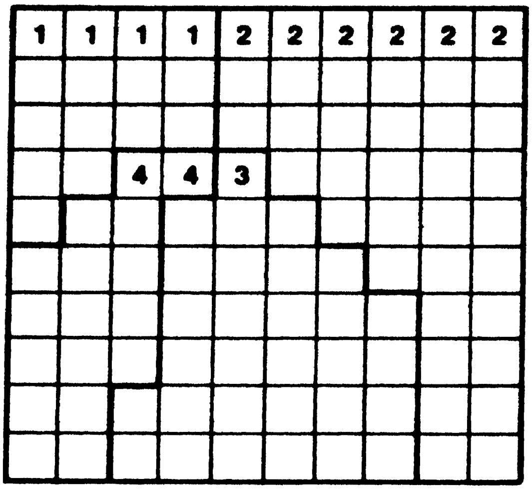


Output

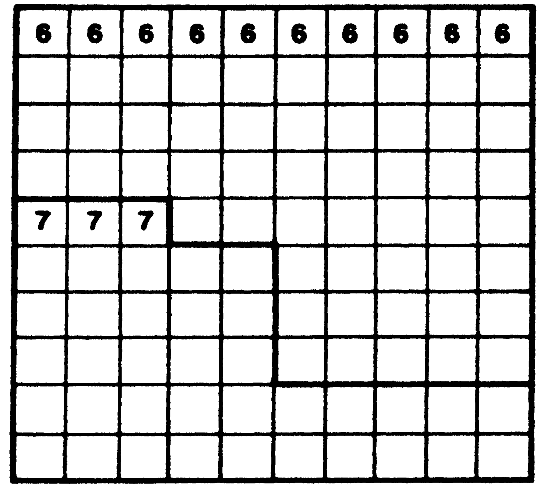
Class 4 is assigned to Class 3

Reclassification Example

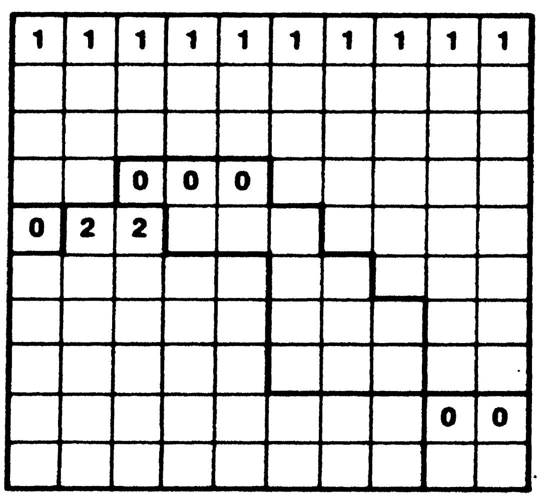
FIGURE 8



Input
Map 1



Input
Map 2



Output Layer

1 is assigned to 1-2 in Map 1 and 6 in Map 2
2 is assigned to 3-4 in Map 1 and 7 in Map 2

Overlay Function Example

- Distance Functions. Distances are used in this class of functions. For instance, a buffer zone around a linear feature may be created, in which all cells that are within a certain distance from that feature are 'turned on'. Distance is a rather flexible concept: it does not always mean straight line distance. It may be the cost along a path, and the path may not be straight. This type of function usually operates within a single map, but at each step multiple cells are involved in the calculation of a single output cell. An example of a distance function, buffer zone generation, is shown in Figure 9.

- Neighbourhood functions. The cells in the neighbourhood of a cells position are used to compute the value for that cell in the output map. Typically, a roving 'window' is passed over the map, and the center cell of the window receives the value computed on the cells in the window. Maps derived from digital terrain models are frequently made with this technique. Another type of map derived from neighbourhood operations is the diversity map, in which the output map is an index of the variability of the input map. An example of a neighbourhood operation is shown in Figure 10.

Display - Results obtained with a raster based system can be displayed in either of two modes: on a display screen, or in hard-copy form. For discrete (such as polygonal) data both types of display have their drawbacks in comparison with vector based systems.

Raster screen displays are frequently limited to 8 bit display values for a maximum range of 0-255. This may be an disadvantage when the range of attribute values considered, such as elevation, goes beyond 255. In this case one may have to resort to look-up tables to map the image of interest to the display. The problem has been solved with the newer 24 bit displays.

Another limiting factor is the size of the image that can be displayed. This size may be as small as 40x40 pixels (GRIDAPPLE), or can be 240x256 pixels (RIPS), or most commonly is 512x512 pixels. Newer displays may be able to hold a 'virtual image' in display memory, that is larger than the physical display area. The user can then access the larger image with built in 'zooming' and 'panning' functions, that operate independently of the host computer. Some displays can operate either in low or high resolution mode, depending a setting of the display controller, but cannot be used in both modes at will. Better displays are able to present an overview of the entire image.

With screen displays, the selection of a single feature of interest sometimes may present a problem. It is possible that one may have to construct an entirely new image, in which all other features are blanked out. However in some display systems, it is possible to assign a new 'mapping' of the image values to the value actually displayed, making the selection of a single feature faster than possible with a vector based system.

Images can be displayed in different colour spaces. Red, blue and green represents the traditional colour space. However increasingly transformations are made to the space of hue, intensity and saturation. In this space interesting displays can be obtained

FIGURE 9

						3	3	3	
					3	2	2	3	
			3	3	2	1	2	3	
		3	2	2	1	2	3		
	3	2	1	1	2	3			
3	2	1	2	2	3				
2	1	2	3	3					
1	2	3							
2	3								
3									

Input Map

Output Map

Distance function example, zone generation. The zone is divided into rings, each cell has value of ring number plus one.

FIGURE 10

1	2	2	3	3	1	1	1	6	5
1	2	3	3	1	1	1	6	6	5
2	2	4	4	2	2	1	7	7	6

Input Map

6	9	11	12	6	4	9	19	22	
7	11	14	10	6	5	15	26	24	

Output Map

Each cell value in the output map is the sum of the cell values in a 2 x 2 sliding window with the output position in the upper left corner.

Neighborhood Function Example

by modulating the intensity according to one attribute and hue and saturation according to others. For instance, for a shaded relief terrain model, hue and intensity may reflect elevation, while intensity varies with reflected light.

It is not easy to obtain map quality hard-copy products with raster based systems, short of a major investment in specialized hardware. Film products can be obtained using cameras specially designed to capture screen images, or one may even used a digital film recorder system. In some systems, the creation of annotation presents a problem, for a lack of an overview of the entire image, and for a lack of appropriate software and fonts. Calibration between screen and film colours is also a perennial problem.

Rather expensive hardware can be purchased to develop paper map products from raster based images. One is a colour ink jet plotter, while another is a dot matrix black and white display device. Both can handle raster and vector data through accompanying front end software.

There are some proponents of raster based processing who see it as the foremost digital procedure for the creation of high quality map products. They are advocates of using an image resolution that defies the detection of individual pixels in the output product. This approach requires the processing of large images (4000x4000). To keep the cost down for this type of effort they advocate the use of rather small CPU's with attached array processors. This is the case at the International Training Center for Aerial Survey at Enschede, where there is a direct link between the raster representation and a lithographic or silk screen process through special software that converts directly from a raster to a screen representation (silk screen printing is far less expensive than lithographic printing).

Another weak side of raster base systems is the lack of a direct link between world coordinates (UTM, latitude and longitude) and display coordinates (line and sample number on the display). This forces the user to make the desired connection at the time he is interested in the display of a map area. This is one reason why most raster based systems belong to the GAS category.

Reporting - A common characteristic of raster based systems, due to the traditional emphasis on image analysis, is the weak connection between the image and corresponding attribute, ancillary and other tabular data. This is the reason for a corresponding lack of a strong report generation capability. Most systems with some GAS capability are able to report pixel counts by category. In some, the user must explicitly deal with the conversion factor to hectares or other areas measures every time such a conversion is required.

Some systems have a link between the image functions, and other table handling and statistics functions. This link mostly takes on the form of a common identifier, for which the user is responsible. He must then exploit this link to produce area summaries by categories.

User Interface - The user interface is one of the important characteristics of a system, because it determines largely how easily the user can state his problem to the system,

and thereby get his goals accomplished, although there are other factors involved, such as speed, reliability, size of problems, output possibilities, etc.

Basically there are four different modes in which a user interface can be cast. They are: menu, question and answer, command, and language.

Menu, as implied gives the user a list of possible choices, from which he can pick one item. Often this is the selection of another sub-menu, that may in turn have other sub-menu, etc. One therefore speaks of a menu tree. The menu mode is typically used when the system writers assume that the basic user is not intimately familiar with the system, or when possible choice are many and complex, or subject to frequent change. An experienced user may become annoyed when he is forced to descend through the menu tree to his function of choice time and time again.

For this reason many systems do not have the menu mode, because the choices are not too numerous, and it is assumed that the potential user can peruse the documentation to become quickly familiar with those choices. A number of systems therefore only have command mode. Others, to accomodate both beginning ands experienced users, have both modes.

The choice between command mode or a language is an interesting one. In command mode, each command usually requires a number of parameters or command modifiers. They have to be entered in a certain order, often in conjunction with keywords or special symbols. An example of a command with parameters and keywords is the MULTIPLY function mentioned earlier. A properly constructed command for multiplying three maps using this function is:

MULTIPLY mapone VERSUS maptwo VERSUS mapthree FOR mapmult

Here, mapone, maptwo and mapthree are the input maps, while mapmult is the output map. In this case the user has to remember the command MULTIPLY and the keywords: VERSUS and FOR, and their correct order.

The rules for constructing a command 'phrase' are called the syntax. The nature of the command syntax for a number of systems is such that the syntax is rather restricted and limited in order to keep the command simple. One must pay for this simplicity with a larger number of commands to allow the user to express himself. The trade-off is therefore the flexibility of the command syntax versus the number of commands.

An example of this trade-off is a user language that was developed by the author that can be used to replace all the reclassification and overlay functions mentioned earlier (cell by cell functions). For example to multiply three maps in this language one would simply say:

mapmult = mapone * maptwo * mapthree

To add in this language one would simply replace the multiplication symbol with an addition symbol.

To perform the CROSS example mentioned earlier, one would simply enter the following statements:

```
mapcross =
if mapone in (1-50) and maptwo in (30-80) then 1
else
if mapone in (50-100) and maptwo in (0-30) then 2
else 0
```

Using the MAPS CROSS command the equivalent command phrase would be:

```
CROSS mapone WITH maptwo
ASSIGNING 1 TO 1 THROUGH 50 AND 30 THROUGH 80
ASSIGNING 2 TO 50 THROUGH 100 AND 0 THROUGH 30
FOR mapcross
```

Note that in this phrase the default alternative (when the conditions are not met) is not explicitly stated, while the 'else 0' in the former example covers all possible conditions.

It is the authors contention that the more flexible syntax and the reduction in the number of commands give the user a significant advantage, while the programming like syntax provides a concise expression as to what is accomplished with each use of the function. However, not too many systems adhere to this philosophy, while those that do may have an awkward syntax.

Systems On Mini And Larger Computers - The distinction between the low end of the mini scale and the micro computer is becoming increasingly vague, but one criterion that can be applied is whether a system is upward compatible with a larger machine. If this is the case, for this category, we will assign the system to the mini and larger class.

A brief description of a systems and its functionality as a IAS, GAS, or GIS is presented in Appendix B for systems of the following vendors: Dipix,(ESRI GRID), ERDAS, International Imaging Systems.

Systems On Microcomputers - Systems on microcomputers are currently proliferating at a rapid rate, and they are becoming more realistic with increasing capacity. New systems are emerging that are based on the IBM PC XT and the newer more powerful IBM PC AT personal computers. One interesting new development is Specdat's VIP processor. With this device the image display functions have been separated from the main computer, and the buyer is free to select from a number of personal computers, including Cromemco, Apple and IBM PC.

Systems from the following vendors are described in Appendix B: ESRI (GRIDAPPLE), SPEC-DAT, and Swedish Space Corporation.

Vector Based Systems

The organization of this section is similar to the one describing raster based systems. In contrast, however, vector based systems appear to be far more diverse in structure and user interface than raster based ones. Raster systems are 'function' oriented, but vector systems may be more integrated and not be as modular.A typical set of functions is not easily defined, and therefore we will place more emphasis on system characteristics. There is a large category of vector systems that is not of interest in the forestry context, such as the Computer Automated Drafting/ Mapping Systems (CAD/CAM). We will therefore limit ourselves to systems that have been used for forestry related purposes. They have also been called 'forestry map information systems.' This term might seemingly imply a narrow and restricted use of a specialized system. In reality a number of unique and different application areas exist within the general field of forestry.

Some application areas identified by the author are: 1) periodic forest inventory, 2) day-to-day tracking of forest land, and management activities, 3) assessing regional timber supply situations; 4) developing and tracking of harvesting plans, 5) short term planning and management optimization, 6) long term planning and optimization, 7) forest pest control, 8) wildlife habitat evaluation, and 9) fire management.

Illustrations of some of these activities are found in the following applications of forest map information systems.

A timber company with landholdings in northern California and the state of Washington uses a system to formulate and track harvesting plans. A company developed 'networking' and optimization program is used to identify harvestable areas. The areas are then allocated in compliance with strict California harvesting regulations (for instance, adjacent areas cannot be harvested within the same time period).

A consulting firm in northern California is using a system to guide a forest inventory for a large landholding in northern California. Photo interpretations are digitized and entered into the system, as is an ownership map. The report generator is used to stratify stands in stratum lists by ownership; the lists are sorted by aerial volume estimates obtained from photo interpretation codes. Sample stands are then selected from these lists and maps are drawn showing the sample stands in relation to the road and stream network. Plot spacings are indicated on the maps by cross-hatching the stands at the correct interval for the desired sampling intensity.

A large timber company on the southeastern U.S. is currently using a system to build a data base, not only of the company ownership, but also of all other ownerships within supply distances of its mills. A large area is covered, and hence the company has taken a strong interest in remote sensing. A major objective for the system is to provide the mill operations with regional supply estimates. A second objective is to track the companies land with detailed inventory records and updates.

Another large timber firm with operations in the pacific northwest has purchased several systems for use at the forest region level. Ownership, timberstands, streams, roads, topography, and wildlife habitats are digitized. The systems are to be used for day-to-day as well as long range planning. The company is interfacing the system with a

variety of models, such as tree and growth simulators, and macro- and micro-management optimizers.

Recently the U.S. Forest Service has used a forest map information system to map and track insect infestation in the state of Colorado. Map outputs were linked to infestation data stored in commercial data base system (System 2000) and then processed through a pest control modeling program. A private consulting firm in California is using the same system to support forest mapping project work.

Another consulting forestry-mapping firm in the northeastern U.S. uses a system to compile forest base maps with attached attributes. The ability to produce color map displays cheaply is one of the advantages of the forest map information system exploited by this company (Teichholz and Kilburn, 1983).

A Canadian province is currently building a data base of approximately 14 million acres, digitizing forest types, ownership, roads, streams and right-of-ways from interpreted 1:10,000-scale orthophoto maps. The objective is to complete the digitization of 1900 map sheets by 1986.

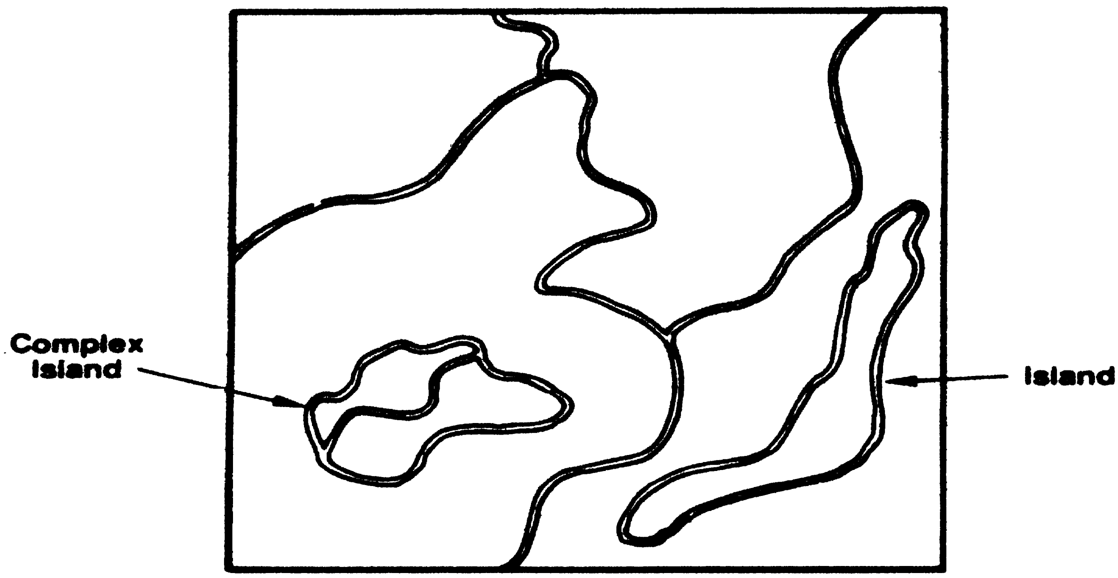
Data Organization - All systems classify map objects as points, line or polygons at some stage. However, data structure and processing methods differ. There are two major types of vector data structures: polygon and arc-node. With a polygonal system entire boundaries of closed figures are stored. With an arc-node structure polygons are decomposed into arcs at nodes (typically places where three or more lines come together), and the arcs and nodes are the basic storage units. When using arcs, one can cope with arbitrarily large polygons, while reducing storage requirements by avoiding double boundaries (however, bookkeeping and attribute link data storage increase). Polygons on the other hand require less bookkeeping and facilitate query processing. Arc based systems are more suitable for handling adjacency information. The current trend seems to favor arc based systems. Examples of a polygonal and an arc-node data organization are shown in Figure 11.

Most systems employ some form of map storage unit, referred to as a 'control unit,' 'tile,' 'map block,' 'geoblock,' etc. One should be concerned about the maximum size of these blocks. Other systems do not divide the area of interest into blocks but store the data as one giant map, on which 'virtual' maps can be defined

Map layers can be subordinated to map blocks or vice-versa. Each mode of organization has it subtle advantages and disadvantages. There may be a restriction on the maximum number of map layers and on the number of (polygons, arcs) that may be stored per layer. In addition, individual analysis functions may have conflicting size restrictions.

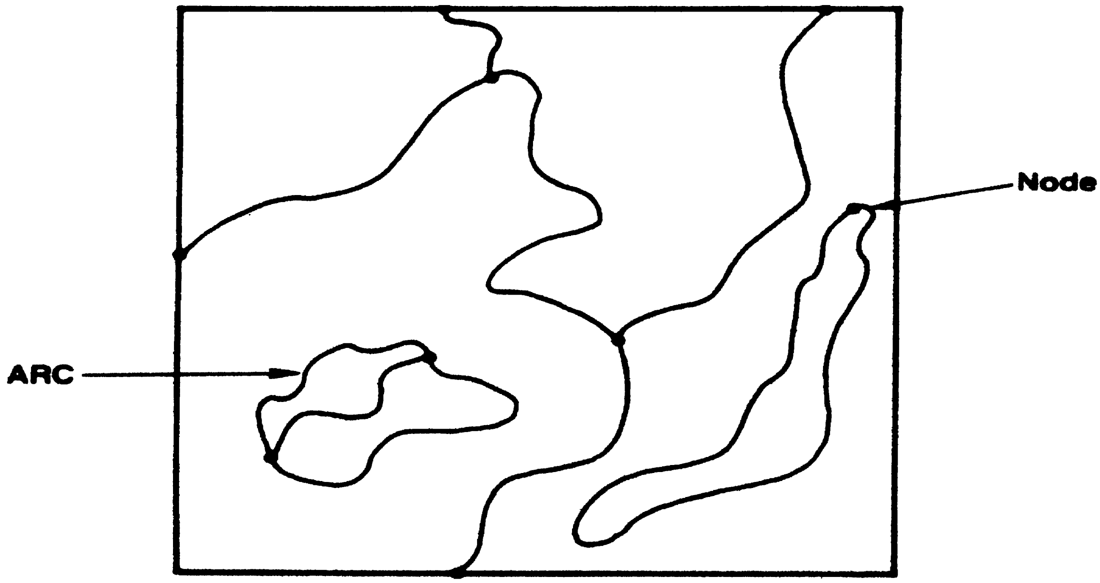
Attribute data can either be interleaved with the map data, or the data can be stored in completely distinct files. An advantage of interleaving may be quick simultaneous access to both map and attribute data with the assurance that both types of data are maintained in a consistent state. Traditionally, concurrent updating of two separate files has proven to be more hazardous.

FIGURE 11



(Coincident Lines Drawn Slightly Separated)

Polygonal Data Structure



ARC-Node Data Structure

Two Vector Data Structures

Different systems have different search strategies, but in general some kind of keyed access helps speed the search for map and attribute data. One system has a special hardware disk scanner to support fast data access.

Database Functions - The database side of a system is featured most prominently on the input side, but the database and data organization are all important throughout the system. The database system forms the backbone of the operation and as much of the quality capacity and security of the system depends on it.

It is therefore important to ask whether the system ties into a commercially developed and sold database system or whether this system has been developed by the vendor. One should be aware of the difference between a 'database system' and a system using the operating system file management facility. Real database systems maintain all data in a consistent state, irregardless of breakdowns, abnormal program termination, hardware failures, etc. They usually have an associated data definition language and employ a 'schema' describing the relationships between data entities.

The concept of a workfile is significant. A workfile is a temporary database or file to which maps or portions of maps are copied for further processing. Data may also initially be gathered in a workfile before being copied to a database. After processing, only good and permanent results are transferred back to the database. In this way the primary data are protected because they are one level removed from ongoing and analysis and input operations.

One must be aware of the maximum number of users allowed simultaneous access to the database and workfile. This is different from simultaneous access to files managed by the operating system. Most systems do not allow simultaneous access to the same area, but instead use a 'lock-out' mechanism to prohibit concurrent use

Input Of Map Data - One form of input of map data is through manual digitizing. Strong differences of opinion exists for this type of map input. One controversy is that of 'blind' vs. 'interactive' digitizing. With the blind approach there is no continuously updated display of the work in progress. Depending on ones viewpoint, the operator is either distracted or aided by viewing a display with the on-going work. Some systems have a blind as well as an interactive digitizing mode, while others are completely interactive.

Another controversy is whether to digitize in either 'stream' or 'point mode.' In stream mode the cursor is continuously sampled at a set time rate or distance interval. Most operators prefer point mode because they can relax between points.

All systems have some form of interactive digitizing. In the ideal case, the user only has to edit those situations where the current input does not correctly match the map. However, in most systems the user also gets involved somewhat in defining the maps topology. Topology refers to the relationships between the basic elements stored in the spatial database. For instance, when the unit is a polygon, an island (for example an island in a lake) must be situated inside an outside polygon (the lake boundary). When storing arcs, each arc can only connect to two nodes, and each arc must have a left and right area identifier. Furthermore, arcs cannot suddenly end, or 'dangle', they must

connect to other lines. When constructing areas, interior arcs must be used twice, except when they are part of an island, and arcs at the border of a map should be used only once. With one system the user digitizes line entities and then runs these entities through a 'health checking' phase. This system flags all deficient junctions (nodes) by surrounding them with different symbols for different error types. The user then has to manually edit the errors and return to the health checking program.

In other systems, 'overshoots' and 'dangling' lines are automatically removed, and hence errors are more related to a lack of correspondence with the input map.

All systems have capabilities for editing previously collected data. The user interface consists of either a digitizer board menu or a CRT screen menu with either keyboard or digitizing pad responses.

Most systems require the digitization of a desired label location within a polygon: in some cases this location is also required as a starting point for polygon definition. To define a polygon, the system starts at the label location, finds the nearest arc, follows this arc to the next node, takes the sharpest clockwise turn, and so on, until it arrives back at the starting point. With other systems labels can be placed automatically.

Some systems have the capability to compute lines and polygons directly from surveying data. Some also have a variable resolution capability. When arc data are stored, lines can be 'thinned' or 'weeded' (removal of specified points according to a tolerance). The reduced arcs can be displayed as a map with reduced resolution. However this process works only up to a point, at which entire arcs need to be removed. No system has as yet a satisfactory capability in this area.

Another form of input is through map scanning. Current equipment is still very expensive to allow for individual ownership of this service, and mostly one must resort to a service bureau for this type of work. However, low cost scanning devices are on the horizon. Tektronix has annouced its new Model 4991S1 Graphics Input workstation which is capable of scanning maps of 35x47 inches. This system cost approximately $150,000 but this is a magnitude less than previous systems. The Canadian Land Data System was one of the first GIS systems to employ a map scanner. With a map scanner productivity factors may increase from 5 to 10 fold over manual digitization.

Input Of Attribute Data - A controversy surrounding the input of attribute data is whether to enter the data at digitizing time or in a separate phase. In the latter case, one does not tie up an expensive digitizing system with the entry of alpha-numeric data. Some systems allow simultaneous entry of one feature, others automatically assigns feature ID's for polygons while the user assigns ID's to lines and points. All other attributes are then associated with the initial feature ID. With one system, the entity to which the attribute data must be attached is selected interactively by pointing to it with the cursor.

Most systems have some arrangement to check the validity of input data; it may be less obvious whether the consistency between map and attribute data is checked. One system will not store a map in the database if there is not a one-to-one correspondence between spatial units and attribute data assigned to units. Most systems also have

automatic prompting (or a form that appears on the screen) for input of attribute data, or free form input may be used if the user is familiar with the required input stream.

In considering attribute data, one should note special requirements for this type of data in the forestry context. In the author's experience, of all natural resource disciplines, forestry provides the most quantity and variety of data for storage in a GIS. One should be able to store information such as stand and yield tables, variable length species composition vectors, log grades, etc. This type of information frequently results in variable length records. Some systems allow only fixed length attribute records with a predetermined number of attributes.

Handling of unknown data is another important consideration. Often data are yet uncollected, or they may be irrelevant in certain situations (for example, current annual increment for a water body). Some systems have special codes for these situations, consistent with the retrieval logic for known data.

Finally, it is important to know whether new data can easily be computed from existing data (such as total volume for a unit, when vol/ha is the stored item). In some systems computed data must be stored, while in others they can be computed 'on the fly' when a report or map is generated.

Topology Definition - The linework resulting from the digitizing phase is sometimes referred to as 'spaghetti.' Topology definition is the process of untangling the spaghetti to make a meaningful and consistent map coverage. For instance, in an arc based system, all arcs must be connected to nodes and carry the correct left-right attribute information. The arcs must be connected to form polygons, and polygons must have properly nested islands. An example of 'spaghetti' and a structured maps is shown in Figure 12.

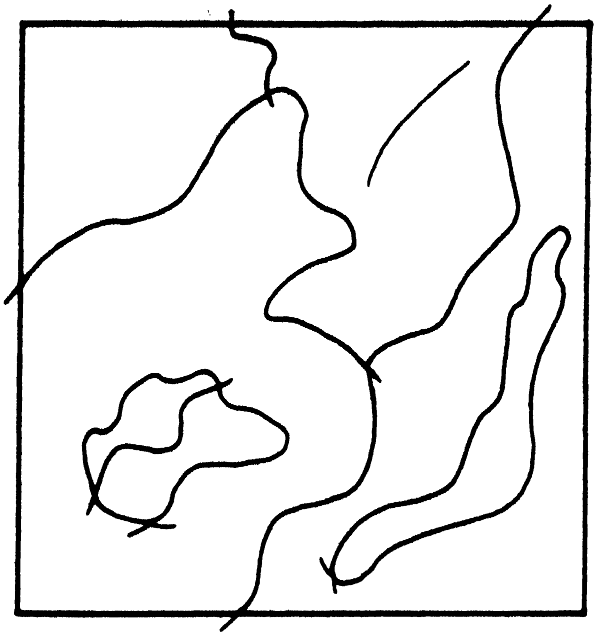
Creating the topology is sometimes called 'complexing.' With one system one has the option to complex either manually or automatically, in future versions the task will probably be automated. Another system has 'Build and Clean' programs, yet another has a 'Polex' polygon creation program, while a third system uses a 'Knots' program to resolve arcs and nodes.

In the past, islands (polygons within polygons) have required special processing in some GIS functions, and one systems still uses a 'nested list file function' to determine which polygon is wholly contained in which other polygon. Most systems today automatically assimilate islands without special user intervention.

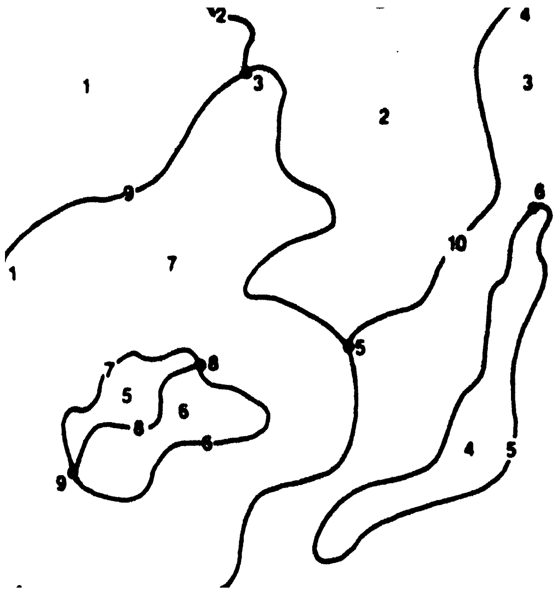
Topology definition may cause some automatic editing. 'Dead ends' or dangling line segments may be removed and polygons that are too small may be automatically merged into larger ones.

Updating Of Map Data - Existing map data can be updated in two ways once a change in map coverage has occurred. With the first method an entire map block is re-entered in the new configuration or at least the changed area, and all lines affected by it are re-entered. With the second approach one only digitizes the changed area and then merges the new area into the old map by performing a 'paste up' updating function which is similar to the

FIGURE 12



"Spaghetti"



Structured Map

Arc Data
(Outside of Map is -1)

ARC	Left Region	Right Region	Start Node	Stop Node
1	-1	1	1	2
2	-1	2	2	4
3	-1	3	4	7
4	-1	7	7	1
5	3	4	6	6
6	7	6	8	9
7	7	5	9	8
8	5	6	9	8
9	1	7	1	3
10	2	3	5	4

Topology Definition

'hidden line removal' encountered in 3-D perspective operations. The merging process also updates attribute data of the old area; surface areas are adjusted and data are duplicated when entities are split. An example of a 'paste-up' update is shown in Figure 13.

Rather than updating an entire map, at times there are occasion when merely outdated attribute data must be changed. One would then like to change only those records pertaining to the changes without having to re-enter other data. Due to the complexities of storing variable length matrices and vectors, this is currently not possible in one system. In other systems one can change individual items, and this capability is enhanced by the ability to indirectly point to a master file with names and definitions. A change in this master list changes all entities pointing to the list.

Ideally, one should be able to edit through a selected database query. The query specifies what attributes should be changed and what they should be changed to, using Boolean and arithmetic expressions in natural language constructs.

Attribute data are frequently organized according to a 'schema.' At times, as when adding a new land property to the database, one might wish to change the schema. It then may be desirable for both the old and new schema to co-exist or else it should be possible to adapt the old data to the new schema.

Selective Retrieval Of Geographic Areas - Designating a certain geographic area and then operating or displaying this area is an important function for input, display, analysis, and query purposes. With the 'workfile-database' concept, the selected area is retrieved from the database and then stored in the workfile for further analysis. An example of a retrieval by geographic area is shown in Figure 14a.

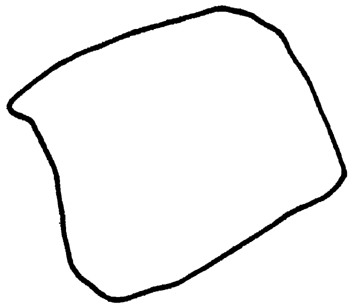
Two basic modes of retrieval exist. If the map data has a hierarchical organization, areas may be requested at each hierarchical level. One system provides for such an organization. If the coverage consists of rectangular units such as map blocks, one should be able to specify the desired area in terms of the relevant map blocks or else the system should be able to determine these blocks from a coordinate description of the area. Map blocks are frequently tracked through a 'catalog system.' The catalog is an index of all map units and their current status (active, archived, etc.)

The border of a combination of map blocks may not always coincide with the desired rectangular area. 'Windowing' may be necessary. Some systems have special functions for windowing, while in others, one must 'overlay' a subset of the database with the rectangular area.

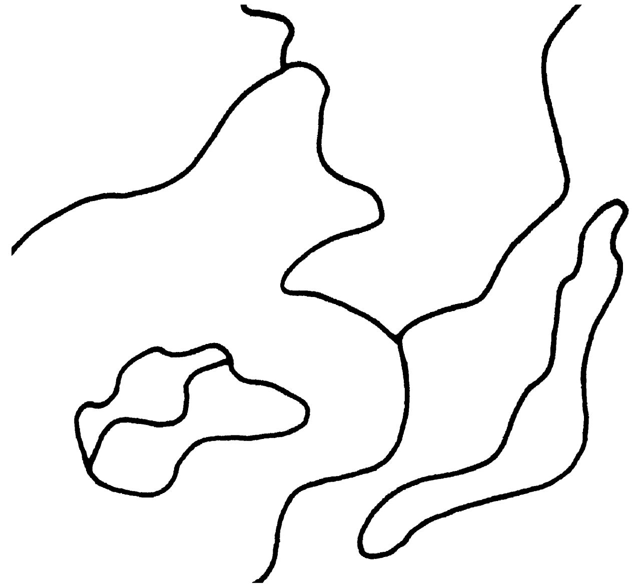
In any case, an overlay function must be used for all systems when an irregular area is required. An exception occurs when this irregular area is the boundary of the union of irregular map blocks with irregular boundaries.

If the windowing operation is performed in real time, a 'zooming' effect may be obtained. Fully interactive systems are particularly effective in this area. A 'scrolling' effect results when the window is moved around in real time; none of the systems

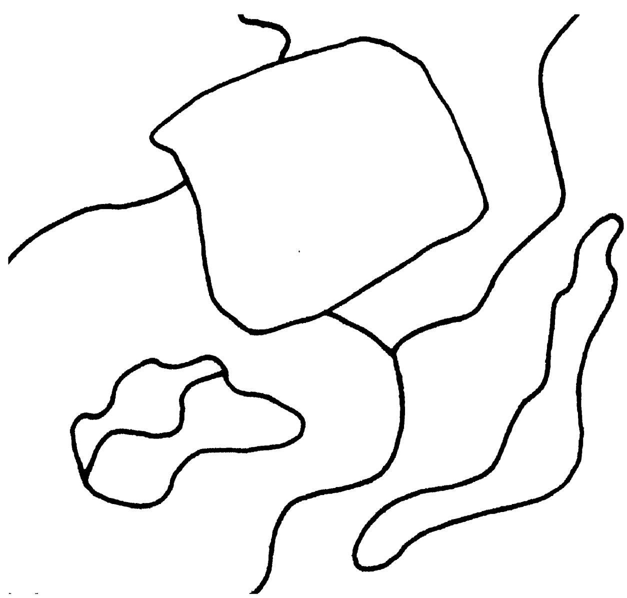
FIGURE 13



New Input



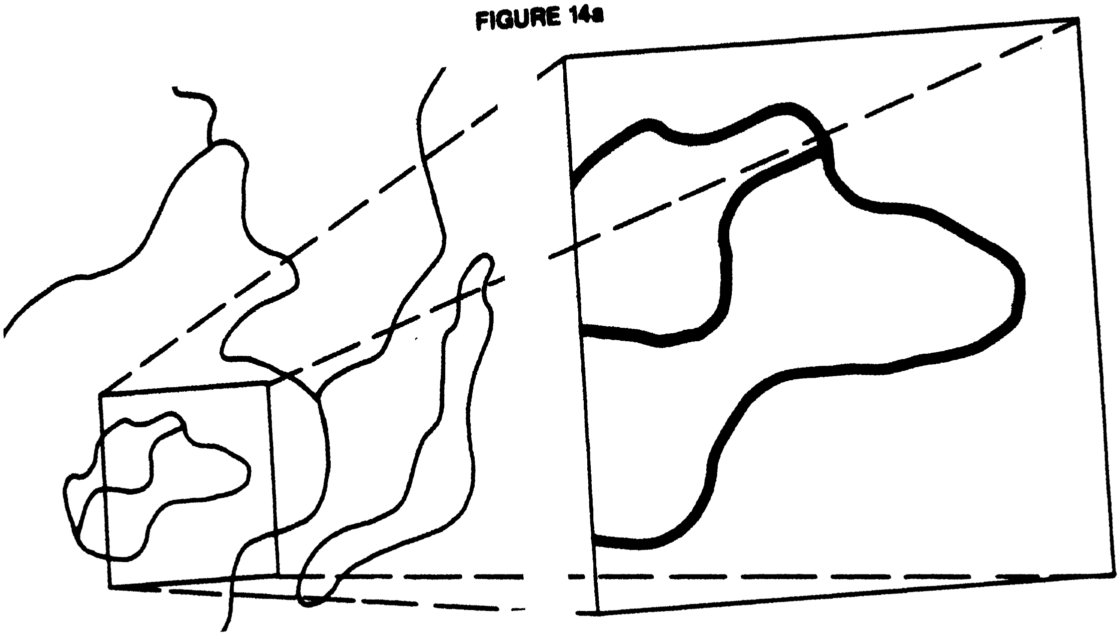
Old Map in System



Updated Map

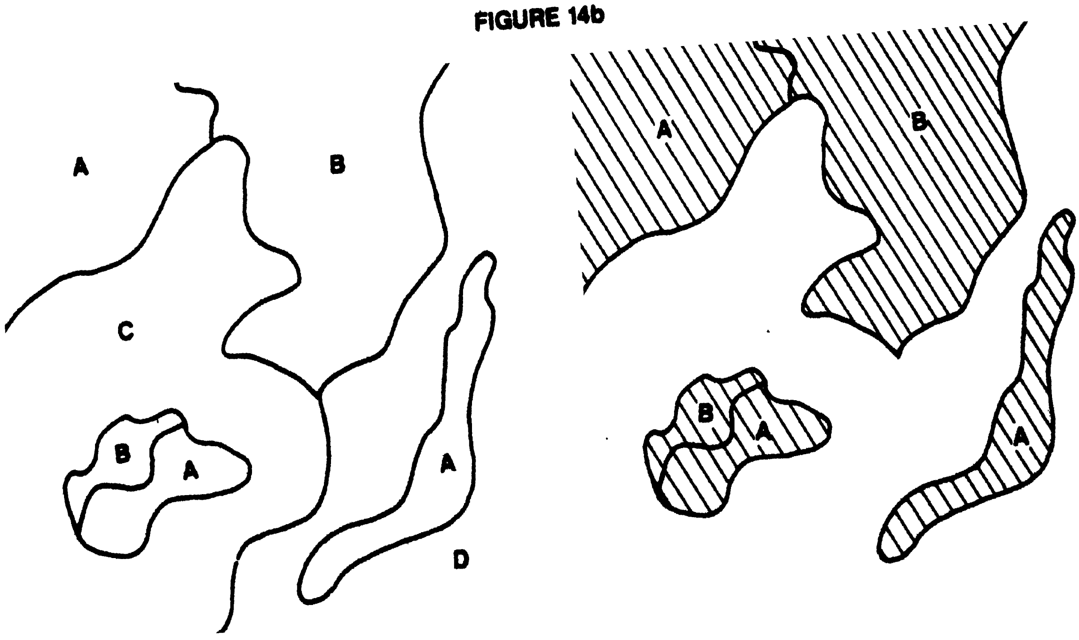
'Paste Up'' Updating

FIGURE 14a



Selective Regrieval by Geographic Area "Zoom"

FIGURE 14b



Selective Retrieval by Attributes "A or B"

presently have this capability.

Selective Retrieval Based On Attribute Data - All systems store spatial entities such as points, lines and polygons. Attribute data are stored with each entity. These may be as simple as a fixed length character string (a'subject') or as complicated as the attribute data associated with an entity in a special attribute database.

Whatever the complexity may be, all systems allow selective retrieval of spatial units based on attribute data. Mostly this function takes the form of specifying a 'Boolean condition' either in equation form or through a set of interactive prompts. For instance, to retrieve all covertypes 'PP' or 'WF' with stocking greater than 128 and origin date before 1/1/1930 one might enter a statement with the following syntax, as part of a query:

```
IN LAYER = COVERTYPE LAYER
STANDTYPE EQ 'PP' OR STANDTYPE EQ 'WF'
AND SITE INDEX GT 128 AND ORIGIN DATE LT 1/1/1930
```

This statement would then return a map of timber stands to which thinning operations could be applied. An example of selective retrieval by attributes is shown in Figure 14b.

Specifying the condition in a 'natural language' equation form seems to be preferred; it is more 'user friendly' than requesting a series of operators and operands with prompts.

Data retrieval is generally speeded up when attributes are designated as search keys or parts of search keys organized as an index. Frequently used information such as item number, layer identifier, and entry date are part of the record key in one system. Other systems may also have the capability to key on attribute data.

A recurring issue is the handling of unknown data. What happens when an 'unknown' attribute is encountered in the search? A system must have a strategy because unknown data will occur.

An overall coherent approach to feature selection is obtained when the attribute 'filtering' statements are part of a more comprehensive query facility in which the map combining operations discussed in the next section are also included.

Map Layer Combining - One of the most important and difficult to implement features of a forest map information system is the facility for combining different map layers. This function is mostly referred to as map overlay, although there are more possibilities for this computerized function than a physical overlaying of two maps would indicate.

Usually a system deals with either point, line or polygon layers. Theoretically one can therefore make six different type combinations, but the most commonly encountered are points on polygons, lines on polygons, and polygons on polygons. For instance, to locate inventory plots within a certain forest type, one performs a point on polygon operation, whereas to compute the number of kilometers of a road type within a forest type, one must use a line on polygon operation. When a system has a polygon on polygon capability only, line and point functions can still be simulated by creating small buffer

areas around points and lines.

The most common functions for polygon to polygon combinations is the intersection (AND) function. All systems have this capability. Not quite as common are the union function (OR) and complement (NOT). Only two systems known to the author have AND, OR, and NOT functions. One system prefers to tailor its systems overlay capability directly to the users requirements. Examples of AND, OR, and NOT functions are shown in figures 15, 16, and 17.

The update capability referred to earlier is directly related to the overlay function. If map A is to be supplemented on map B for an update, then the update function can be thought of as a composite overlay as in the following form: (A OR (B NOT A)). A genuine GIS (not GAS) system should have an implementation of this updating function to keep the stored area up to date in relation to the real world.

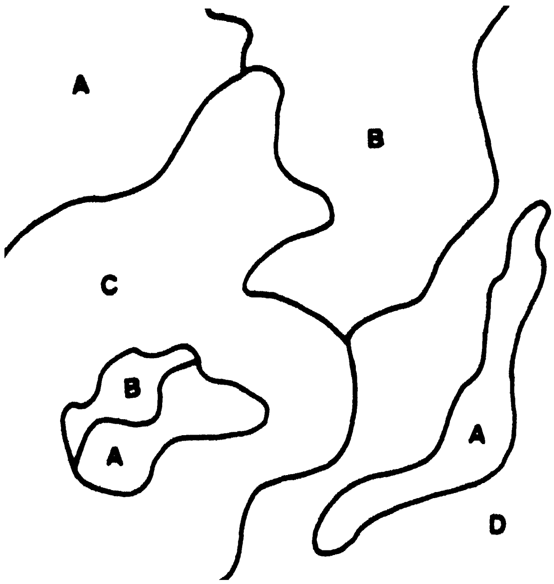
Performing a union operation within a map layer is sometimes referred to a a 'merge and dissolve' or 'drop-line' operation. Performing the union function but keeping the original boundaries is another possibility. In one system the map combining logical operations are a part of the overall query language. For instance, when requesting all standtypes such that (STANDTYPE EQ 'PP') AND (SOILNAME EQ 'HUGO'), the user does not have to be unduly concerned with the fact that the first and second term in the expression are within map layer entity selection operations, whereas the AND connecting the terms must be executed as a map overlay of selective vegetation and soils layers.

In all systems, combining multiple map layers can be accomplished by first combining two maps and then combining the result with the third and so on. In one system the possibility exists to combine multiple layers in one user transparent pass dictated by the problem expressed in the query language. The user invokes the overlay program only once.

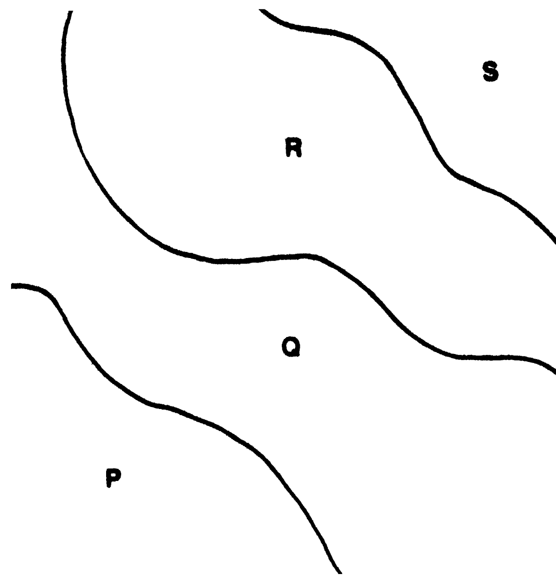
A difficult problem associated with overlay processing is that of slivers. Significant lines on different map layers may deviate slightly and create 'sliver polygons.' The problem can be avoided from the beginning by copying these lines from layer to layer to insure that they are absolutely identical. But this is not always possible, and hence a sliver elimination capability is a useful feature on an overlay program.

One of the most hidden but significant processes occurring when combining maps is the transfer of attribute data to the resulting map. For instance, when intersecting two layers, the polygons in the new layer should acquire the attributes of the underlying polygons of the input layers. If the attributes are stored in a single character string, the strings can be merged to form a new string for the output map. However, with a more advanced attribute handling schema, where multiple attributes may be associated with a spatial entity, a pointer system may be used to attach the schemas and attributes of the input layers to the new layer. Repetition of this process with previous results builds up a 'pointer tree.' This tree provides access to each of the original attribute data for each derived layer, no matter how far removed from the input data.

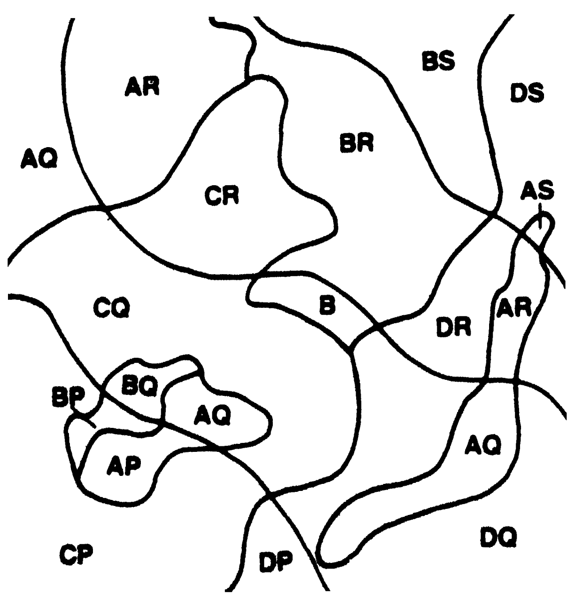
FIGURE 15



Vegetation Map



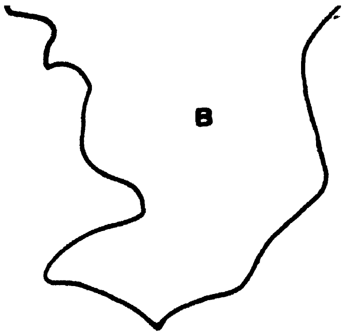
Soils Map

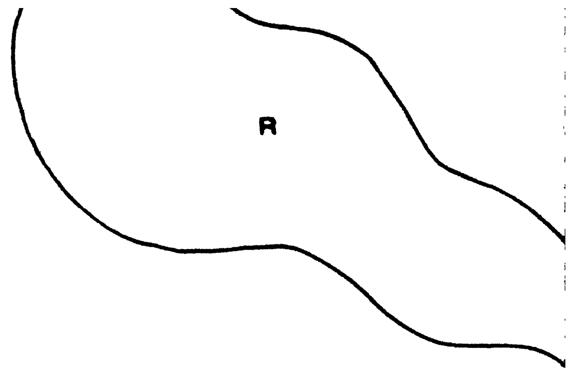


Vegetation AND Soils

Map Layer Combining "Intersection'

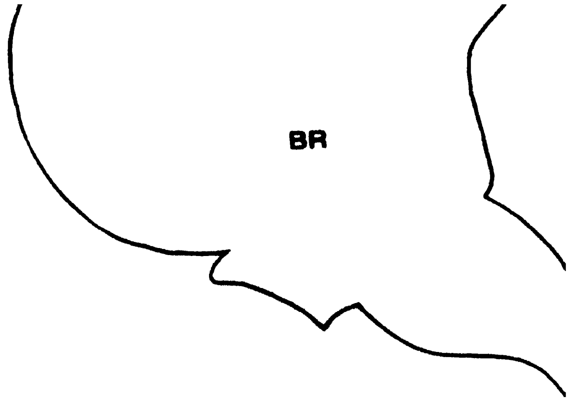
FIGURE 16





Vegetation EQ 'B'

Soils EQ 'R'

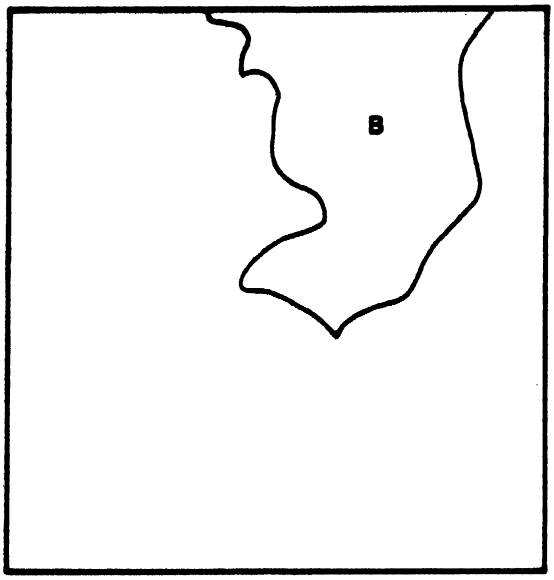


Vegetation EQ 'B' OR Soils EQ 'R'

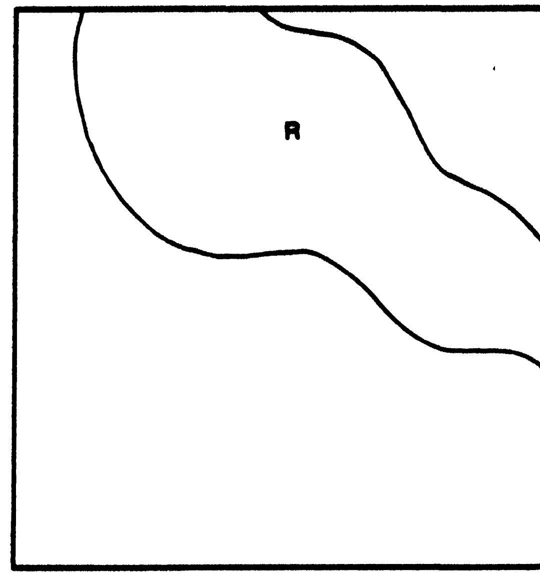
(with 'Delete Interior Lines', also called 'Merge and Dissolve')

Map Layer Combining "Union"

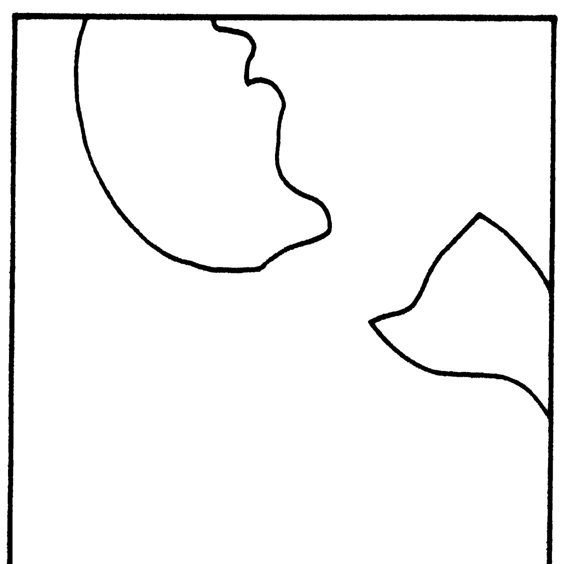
FIGURE 17



Vegetation EQ 'B'



Soils EQ 'R'



(Soils EQ 'R') and (not(Vegetation EQ 'B'))

Map Layer Combining "Difference'

In this context, one problem exists that has not yet been satisfactorily resolved in any of the systems; it is therefore currently beyond the state of the art. It occurs when combining polygons, for instance by dropping a common boundary, which have attributes expressed on a per ha basis, such as vol/ha. For the new unit to have the correct per ha attribute, the overlay process must not only compute the acreage of the new unit but also keep track of the percentage composition of the new unit in terms of the old areas. None of the systems considered has this capability.

Different systems may have drastically differing overlay speeds depending on the type and complexity of the lines and polygons. The CPU requirements depend greatly on the type of algorithm implemented. The magnitude of the differences between systems may be in the order of CPU hours. The only way to assess these performance differences is to benchmark different systems with the same maps, performing identical overlay operations.

Buffer Zone Generation - A variety of forest management problems require that a 'buffer zone' be generated around either a point, line or polygon. All systems therefore have this capability to a degree. An example of a buffer zone generation is shown in Figure 18.

They all can generate a circular area around a point and a 'sausage' like area around a line. With line and line networks a number of problems may occur. Incorrect results may be generated when the line is extremely 'wiggly' and the buffer zone is wide relative to the amplitude of the wiggles. When more than one line is involved or a line crosses back on itself, at least within the zone width, the sausage shapes may overlap, and the resultant zone may have undesirable interior boundaries. Removal of these lines requires the map overlay 'union' function, but the use of such a function is not a common feature.

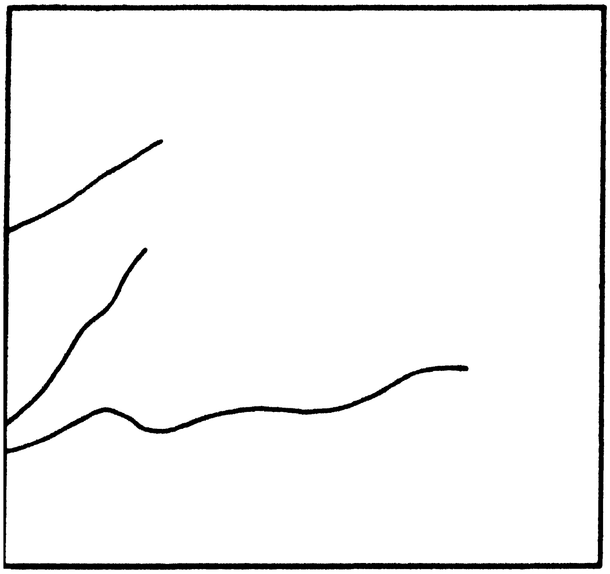
Generating zones around polygons is more involved than generating zones around lines. In one system one can generate 'full zones' and 'half zones.' Half zones can be either on the inside or the outside of the polygon. If they are generated on the outside, then for islands they should be on the inside. One system ignores islands in buffer zone generation.

Terrain Data Handling - There is a considerable variety in the systems capabilities for handling the three dimensional aspect of terrain. All systems can store line data and can therefore handle contour lines.

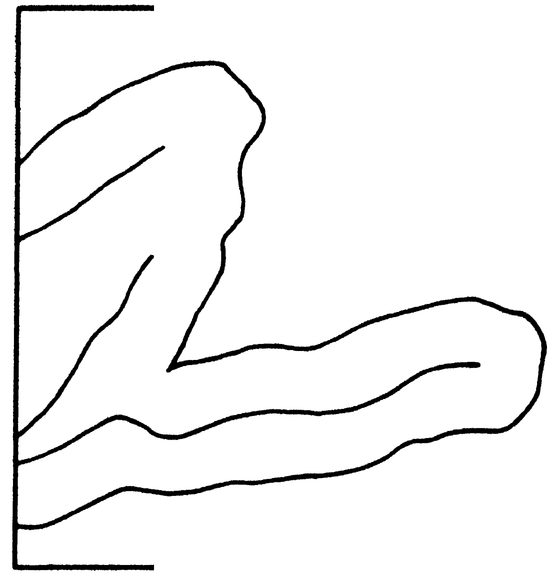
Slope and aspect maps can be generated from the stored contour data. These maps play an important role in forest resource management; slope maps are especially significant. For example, different logging practices may be called for on different soils-slope combinations to minimize erosion hazards. Not all systems have the capability to generate slope and aspect maps from contour data. One system uses a hexagonal raster structure, while others use a square raster structure.

Other capabilities related to the vertical dimension of the terrain are the capability to generate 3-D perspective views, and the possibility to aid in road design work with cut and fill calculations. Many other terrain related types of analyses exists and may be

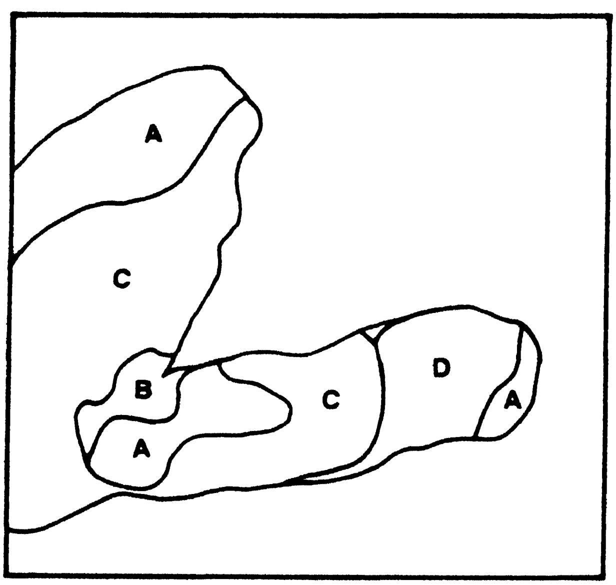
FIGURE 18



Streams



Buffer Zone



Stream Buffer Zone AND Vegetation

Buffer Zone Generation

implemented in some of the systems but are beyond the scope of this report.

Display - All systems have screen oriented interactive display devices on which segments of land covered in the database can be displayed.

Quite often it suffices to review the results of an inquiry on such a device, but more often than not, a more sophisticated hard copy map is required. All systems therefore also have a plotter as a peripheral device in addition to the graphics CRT. Capabilities of the plotting packages are important as quite often an impression of the quality of the work performed by the system is obtained from the quality of the hardcopy plot.

At times different map projections may be called for and so the plotting software may have to transform to a different output projection. For smaller forest properties this is probably not a crucial requirement.

A more important capability is to join individual map blocks. This may be accomplished either by first creating a joint file, or the function may be performed 'on the fly' at plotting time. An important detail is whether map block boundaries can be removed when maps are joined.

Another concern is the quality and ease with which attributes may be displayed in association with point, line and polygon entities. In some systems the attributes can be placed in the polygons automatically, while in others they must be placed interactively. When thousands of polygons must be displayed, this may be an unreasonable approach. Another question is whether more than one attribute can be displayed inside a polygon. Some systems use centroids for label locations. If so, labels may occasionally be placed outside the polygon.

All systems have the capability to display a 'theme' by 'cross-hatching' polygons, but not all systems can cross-hatch and display attribute data at the same time, without drawing cross-hatch lines over the attribute labels.

One should be able to superimpose different map layers. This may lead to conflicts, and the results may be confusing. One remedy is to use the NOT capability of the overlay function to make layer complements which fit together as puzzles pieces. For instance, one may be able to generate a buffer zone around roads, eliminate this zone from the covertype layer, and then jointly plot the result and the road zone using different cross-hatching patterns.

Reporting - Another vital tool to be used in tandem with the map display and map making facilities is the report generator. All systems have a capability to generate reports from stored attribute data. However, ease of use and functionality of the report program may vary greatly.

The report should be indexed to the accompanying maps on a spatial entity basis. Some form of automatic formatting, or a user friendly interface for laying out the required format is important. The program should have a general sorting capability, and it should provide totals and subtotals for certain attributes, such as area and total volume as

indicated by other attributes such as timber type, soil type, or land capability class.

An important issue is also whether the reporting capability is tied to the overlay process and is capable of relating attributes not only to the overlay map unit but also to the original input maps from which the overlay was

Another matter is whether one can report variable size matrices such as standtables and log grade composition. Furthermore, the report should clearly indicate which attribute data are either unknown or irrelevant.

The report generator may also be the formatter for auxiliary outputs, such as required for management planning models.

Hardware - One of the interesting aspects of the systems considered is the great variety of hardware employed. However, all systems have the following main components: Main CPU, disk drives, tape drive(s), printer, digitizer, plotter, graphics CRT, terminals and modems. An example of a typical GIS hardware configuration is shown in Figure 19.

Although a vector based GIS system is certainly not a CAD/CAM system, the current trend towards the use of low cost CAD/CAM systems (Teicholz and Kilburn, 1983) is also present in the evolution of GIS systems. Several firms now present low cost systems developed around micro versions of the larger CPU's used previously. Most other vendors offer 'entry-level' streamlined systems at a lower cost than their regular system. Vendors using upward compatible CPU systems, such as the series of Prime computers, can choose an 'engine' of sufficient power and capacity for a certain application with the entry-level system using the smallest CPU. A given CPU, if equipped with a virtual memory capability, may have different throughputs for different memory configurations. The hardware therefore needs to be specially configured for the intended application to be maximally effective.

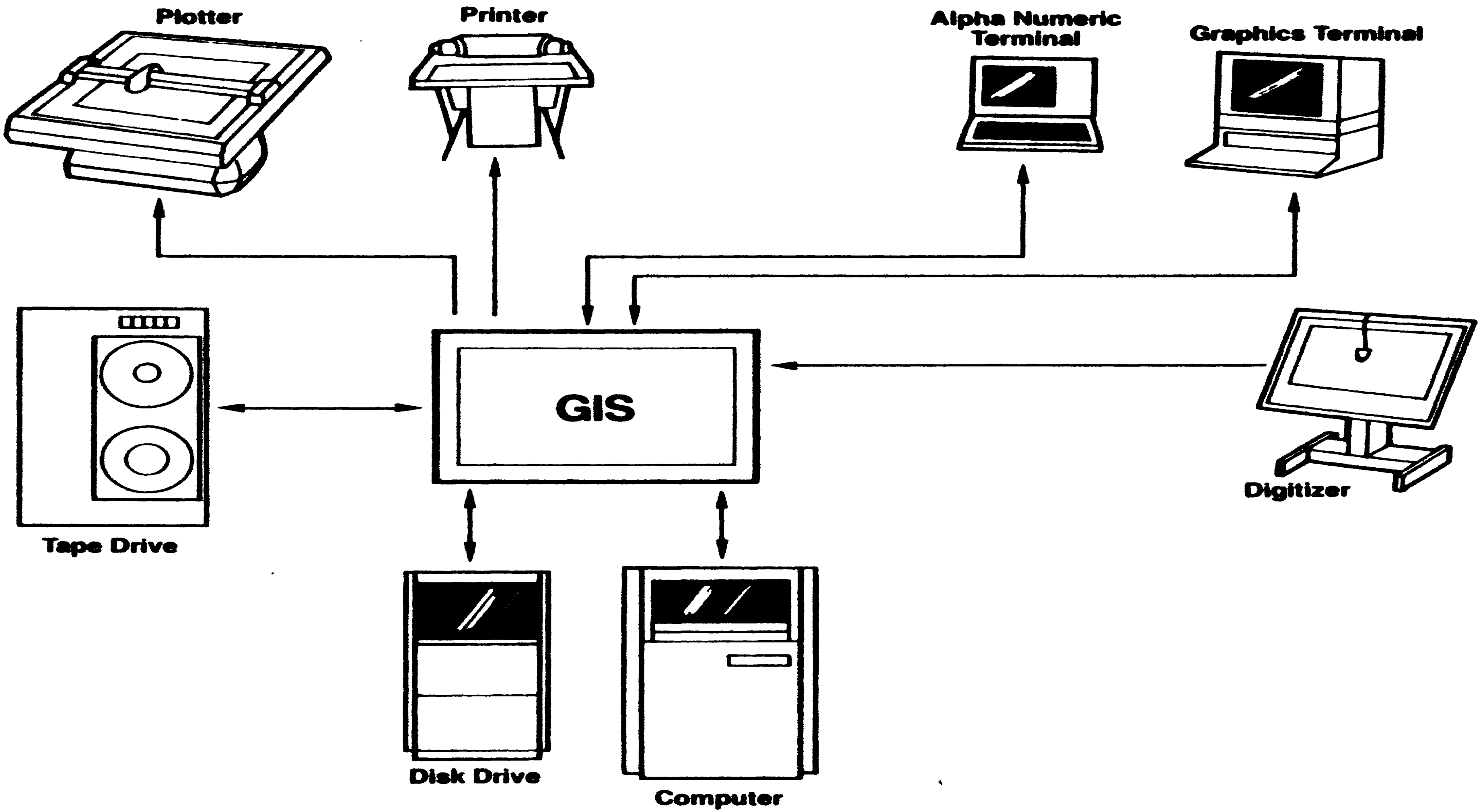
Some systems are more hardware dependent than others. One vendor manufactures a special 'disk-scanner' to support interactive fast retrieval of graphics and attribute data. Some vendors also package the displays and digitizing board into a single integrated workstation.

The current trend for the larger systems is toward the utilization of 32 bit CPU's. Some micro version of larger systems are currently developed around 16 bit CPU's (LSI 11/23), but the long term trend in low cost systems is certainly towards 32 bit processors.

Peripherals can be divided into two main groups: those associated with the main CPU, such as tape drives and disk units; and peripherals to support the spatial data functions, such as digitizers, plotters and graphic CRT's. The more hardware independent systems can easily accommodate devices of different make and model, while other more tightly integrated systems have a well defined composition.

Each CPU can usually support more than one workstation; in some cases smaller CPU's handling different functions are networked together.

FIGURE 19



The GIS Hardware Environment

Current Trends

At present we appear to be at a cross-roads of trends in the development of GAS/GIS technology. Development of a number of systems was begun in the mid and late sixties and early seventies. Some systems were originally designed as vector based systems. Others migrated from the image analysis - raster based world into the GAS/GIS domain.

One of the disadvantages presented earlier for a vector based system is the programming complexity of some of the vector processing tasks. One in particular, the overlay function, is notoriously difficult to program reliably. Similar functions are easily accomplished in the raster domain. The nature of the problem is such that it is not unreasonable to arrive at a solution that will work 99% of the time, but that the approach chosen can never be made to work all the time. Moreover, when a failure occurs, it may be spectacular, and prohibit the further processing of an entire data set. Or, one may be able to fix a problem for one case, but the solution may invalidate another case, that could be processed before. One may then be in a 'push-pull' situation without knowing it.

Realizing the nature of this problem, it is interesting to observe how the frustration level of users of vector based systems has been building to the point where system developments have taken place to cope with the situation. One vendor has scratched his existing vector system, and has developed a completely new system, that is reported to be far more reliable. Before this development a color coded chart was in circulation, indicating the reliability of the various system functions as determined by the users experience.

Another vector based system still has serious problems with some of its functions, especially the overlay operation. In this case the vendor has attempted to solve the problem by having alternate software packages, as available from other systems. If one system does not work, the other may, and between the two systems the chances of completing a problem are reasonable. A U.S. Government Agency that is a heavy user of this system has added a raster based set of functions instead. Overlay processing is easier and faster and more reliable with these raster functions. In general, however, this need not be so. Satisfactory solutions can be obtained in the vector domain and a potential for superior speeds exists, because the data density is far less. One still must keep in mind however that operations on continuous data can only be performed in cell form.

Many GAS/GIS systems that have originated in the raster world now recognize the limitations of cell processing with limited cell resolution, which, once set, can never be improved. They are especially frustrating when considering that the data are captured at full resolution in the vector domain. The existing trend for systems with a raster emphasis has therefore been to integrate more vector functions with the existing raster analysis.

The overall emerging trend seems to be for both raster and vector systems to converge somewhat towards a middle ground, where functions of both types are

available to the system user.

Systems On Mini And Larger Computers

Systems from the following vendors are described in Appendix B: ESRI (ARC/INFO), Autometric, Comarc Systems, Intergraph, Geobased Systems, Forest Data Consultants.

Systems On Microcomputers

An exiting development in this area is the effort of the mini manufacturers to expend the bottom of the line with micro versions of the larger mini systems capable of running the same operating system as the larger systems, and hence providing upward compatibility, so that the GIS systems that once could run on mini computers only, can now run on micro's. The fact that the micro's are not used as timeshared machines makes up for the difference in performance in many cases. Examples are the Data General 20 micro system and the DEC MicroVAX.

In Appendix B, microbased systems from the following vendors are described: Autometric, Comarc Systems, Intergraph.

2.2 Forestry Applied Technology

2.2.1 Inventory Data Processing

The technology described in chapter 2 has found its way into forestry, because it is general and can be used without any changes or with limited modifications. Thus, the forestry sector has profited from developments which have required enormous capital investments, but which have been economical because of the size of the market place.

For technology that is peculiar to forestry, this has not been so. In many instances, there is an insufficient number of potential buyers to warrant the development of a commercial package or turnkey system solely for forestry applications. A case in point are inventory data processing packages. They are generally not commercially available.

There is another reason for this lack of availability however: the diversity of designs and methodologies used. Wilson, Schreuder, and Kent (1981) make the following statement: 'the sampling design chosen is dependent on the situation at hand, and so also will be the method of calculating the related estimates. Consequently, inventory specialists are forced to either use those sampling strategies for which they have data analysis software available or to write their own software for those strategies for which they lack approriate software. The latter has been chosen more often.' Space (1978) notes that 'the customary method of processing forest inventory data has been to write a custom written program for each application.'

This is not a very desirable situation. One must have a thorough statistical

background, as well as be versed in statistical programming techniques to implement the appropriate estimation formulas and to avoid pitfalls such as roundoff errors. One must also have the practical programming skills to cope with data editing, data sorting as well as data storage and retrieval. To solve the problem, some institutions have tried to fill the gap for their constituents by writing their own generally applicable package. Most notable is the Food and Agriculture Organization of the UN with its FIDAPS system. Another example is the development of FINSYS by the USDA Forest Service in the US.

A different solution mentioned by Wilson, Schreuder, and Kent (1981) is to apply a survey sampling package primarily developed for statistical applications other than forestry. They describe the following: OSIRIS IV, STDERR, CLUSTERS, SUPER CARP, Causey program, and PASS.

2.2.1.1 Systems On Mini And Larger Computers

FIDAPS

This package has been under development by FAO for some time. It has been designed to process data for a variety of sampling designs, and to provide for a general capability to perform data preparation and editing. FIDAPS processing is performed in four phases:1) data preparation; 2) data editing; 3) volume calculations and data grouping; 4) computation of means and sampling errors; and 5) final tabulation.

The statistical formulations for FIDAPS were developed by the Forest Resources Division of FAO. The package was written in FORTRAN 66, and requires less than 64 Kilobytes. Computers on which FIDAPS is currently operational are: IBM 360/370, DEC PDP, VAX, CDC Cyber, and Honeywell.

The following FIDAPS documentation is available at FAO Rome, and can be obtained on request, free of cost: 1) FIDAPS: Application Guide, and 2) FIDAPS: Programmers guide.

FINSYS

FINSYS was developed initially by the USDA Forest Service's Northeastern Forest Experiment Station. It was modified by W.E. Frayer at Colorado State University, and further modified at the Intermountain Forest and Range Experiment Station. The most recent revision of the program is named FINSYS-2. The system can process data for the following designs: simple random sampling, stratified sampling with strata of known or estimated size, unequal probability sampling. It can compute totals, means and ratios, as well as variances, standard errors and covariances.

FINSYS-2 consists of three subsystems: EDIT-2, TABLE-2, and OUTPUT-2.

EDIT-2 is an independent editing and file-updating system. TABLE-2 produces sample summary output of means and variances which is intended for use as input to

OUTPUT-2. OUTPUT-2 then expands to the population level and produces labeled estimates of statistics for the sampled populations.

FINSYS-2 operates on the IBM 360/370, CDC Cyber, Univac 1100 systems and most other larger computers. The system has been written in standard FORTRAN 66.

Wilson, Schreuder, and Kent report that the notation used for FINSYS is somewhat awkward, and that there is a lack of explanation of the used statistical procedures. Schreuder has found errors in the computing formulas for variable probability sampling given in the manual.

3 APPLICATIONS PROBLEMS IN DEVELOPING COUNTRIES

In this chapter we will attempt to address application problems in developing countries. In order to do this, we must examine computer applications problems in general. In the author's opinion developing countries do have unique problems, but mostly the pitfalls and complexities are not basically different from those elsewhere: there is merely a different emphasis due to the development situation.

This chapter will therefore be divided into two parts: the first dedicated to general applications problems and the second providing special emphasis for the situation occurring in developing countries. For the general discussion we will follow some common sense ideas exposed by Covvey and McAlister in their book 'Computer Choices' (1982).

3.1 General Problems

The question to be answered is 'what are the major problems encountered in successfully acquiring, installing and operating a computer system.' On an even more basic level Covvey and McAlister pose the question 'who is the cause for the problems encountered in computerized opperations', and the answer they give is that 'the enemy is us.' They point out that it is our perception of computers and the way we choose to use them that must be feared. If computers are viewed as conspicuous status symbols than money will be wasted on 'conspicuous computing'. They define this form of computing as 'an irrational lust for the aura of sophistication and progress that a person, department, or institution can acquire by becoming computerized.' At the same time there appears to be a growing tendency for us to assume that tasks performed with the aid of a computer can not be performed in any other way. Thus, the need to use computers becomes a self-fulfilling prophesy.

The computer therefore must be put in its proper place. It is a link in a chain of otherwise human processes, or a tool in a toolbox. First and foremost the computer is there to serve us, we must specifically avoid serving computers.

It is true that the computer opens up new vista's of formerly unaccessible terrain, where new things can be accomplished that were formerly unthinkable. In forestry, with its many scientific disciplines, this makes the computer a very exciting tool. But there is the danger coupled with this execitement, of loosing ones perspective, and replacing the goal at the center, with a tool belonging to the periphery.

But the possibility of being left behind in the technological evolution, thereby missing the great benefits that may be could be gained, also exists, and the purpose of this report is therefore to set some guidelines for a suitable middle ground. Clearly the most important weapon in this struggle is to be informed, because with familiarity the computer looses its appeal as a status symbol, and thus allows one to evaluate computer systems and requirements on a practical level. This allows one to demand

solid technical standards of performance from those who develop, sell and implement computer systems.

Returning to the question of the identification of the major problem areas, following Covvey and McAlister, the issues belong to three major categories: consumer issues, technical issues, and management issues, with significant overlap between these categories. They make the point that anyone who becomes involved in computing, sooner or later, will have to convince the people 'upstairs' to provide money to support their plans. These must then be convinced that all problematic issues at stake in the proposed projects have been duly considered, and that there are strategies for anticipating problems that may occur. Even when the people 'upstairs' are computer illiterate, or do not exist, one still cannot ignore these issues because white elephants will be there for all to see, and as a consequence, financially one will soon be left out in the cold.

3.1.1 Consumer Issues

There exists a folklore of computer horror tales. Take for instance the secretary, who mistook a computer for a copying machine and inserted a piece of paper into a vital air conditioning slot, causing overheating of the central processing unit, and thereby the demise of an entire computer system. Or consider (part of the authors experience) the repair man who assumed that a fatal disk head crask had occurred and reinitialized a disk, thereby eliminating weeks of computing results, including all accounting information for more than 10 customers. In the same installation many problems were always blamed on voltage fluctuations, and so an voltage regulator was installed. A fire then promptly broke out in this piece of machinery. This lead to additional precautions, including the purchase of a fire extinguisher. Or take, (likewise seen by the author) the developing country where a voltage regulator was requested for a small Hewlett Packard desktop computer, and a giant crate was delivered, complete with crane hooks.

Many such tales of this sort can be told, and likewise there are many stories of computer systems purchased, but found unsuitable for the job at hand. For instance, a computer may be benchmarked against other systems for speed and maybe acquired on that basis, only to discover later that the operating system is unreliable, halting the machine at embarassing moments. At that point is is to late to realize that a stalled computer has absolutely no speed at all.

As Covvey and McAlister point out, when a system fails, the original opponents take it as proof that their original objections were well founded. With one bad experience, the users may then generalize and conclude that all automated procedures are bad, vigorously resisting any further efforts, while automated procedures are desparately needed.

Unfortunately therefore, the first experience with a computerized application may be critical, but comes at a point when there is the least amount of experience. A second chance may be hard to come by. Not only are the psychological stakes high, but the financial stakes are also considerable. Such a failure must therefore be

avoided at all cost, through proper planning and education. One must know what is needed and wanted, and what one is getting into. However, more often than not, this is not the case, and a leap into space results. The primary cause for such action is what Covvey and McAlister call the 'conspicuous computing syndrome'. This major pitfall has three phases: 1) fascination with computer technology, and a preoccupation with the mystique surrounding the computer; 2) a belief that the computer will provide new powers and elevate ordinary actions in extraordinary ones, and 3) a leap into the unknown based on claims made by compute suppliers.

The fascination with computer technology comes about because of the extraordinary characteristics of the computer. It is extremely complicated, extraordinarily fast, and increasingly affordable. It is being credited with the introduction of a completely new era in the development of mankind, that of the information society. No wonder then, that one may come to think of this extraordinary tool as capable of solving extraordinary problems, but as with most powerful medicines, the application requires a great amount of skill and knowledge.

Computers are seen as a means for increasing the power and the prestige of the institution and its managers because of their public image. They are on the cutting edge of technology and hence any project using computers maybe transformed from the ordinary into something daring and innovative, admired by all. Buying a computer system is therefore a different experience than for instance bying a truck. A computer has associated power, but the truck does not. This aura of power underlies many a struggle for centralized versus decentralized computing. Established computer centers in many organizations are reluctant to give up control to other users who wish to regulate their own computing by using personal or minicomputers. Fortunately this situation is now changing because of the widespread availability and affordability of small computing systems. Computers have an aura of power because they truly can be very powerful devices. By providing appropriate information they can perform feats and lead to insight that is not available to others. Power can corrupt, and at least this can be observed for personal use of computers. The machine obeys command blindly, and hence gives the user a feeling of power. A 'hacker' is an addicted user who may program for a solid 30 hours at a time before 'crashing' next to his machine. At another level people may be corrupted by extreme fascination with the tool itself, becoming obsessed with programming the machine, making the application for which the computer was purchased a minor secondary goal.

Finally, because of a fascination with the technology, and the aura of power that beckons the customer, the equipment is purchased on faith wihtout clearly established goals, without a realization of the utility of the equipment, not knowing whether it will fit the application. It is true that a certain category of systems is designated as 'turnkey' systems. Smaller systems in this category are word processing systems, accounting systems, etc.. These systems are like instruments with clearly defined defined functions. However, the image processing systems and geographic information systems described in this document are far more complex, consisting of sub-components that maybe arranged in a number of ways to create custom versions that are quite unique. The danger is twofold: first, the general system may not fit the users specific needs, an getting the supplier to make the modifications may be very

difficult. Secondly, some systems are put together almost like parts in a kit, computer from one source, software from another, custom peripherals from a third, and soforth. If the total system does not work, the suppliers of the individual parts will not take the responsibility for the system integration, and the customer may be left 'up the creek'. To perform a system integration oneself one may need the tools of the professional system integrator such as analysts, programmers, hardware specialists, etc. Obviously, it may be better to obtain professional help.

3.1.2 Technical Issues

To avoid the leap into space, one must face the technical and management issues of acquiring and operating a computerized system. We will discuss these issues in some detail in the applications guidelines chapter (chapter 4), where both problems and solutions will be mentioned. The most important technical issues are:

- the feasibility study,
- assessing project and user needs,
- developing functional requirements,
- requests for proposals and contracts,
- software engineering
- human engineering,
- privacy and security.
- system maintenance
- database design

Avoidance of any of these issues constitutes a major pitfall.

3.1.3 Management Issues

Similarly management issues that must be considered are:

- the economics of computing
- the risks of computing
- management of projects and equipment
- training of personnel

Obviously, there may be other topics that are just as important or more important,

depending on the situation. It will simply be impossible to deal with all issues in depth, because the field is enormous. Initial additonal reading can be found in Brooks' 'The Mythical Man-month' (1975), Metzger's 'Managing a Programming Project' (1973), and Covvey and McAlister's 'Computer Choices' (1982).

3.2 Problems With Particular Reference To Developing Countries

Without doubt the problems mentioned in the previous section also occur in developing countries. Probably even more so, as the two factors that contribute to the leap into space may be magnified in the development situation. The wider gap between an existing situation and the high technology applied in more developed countries may make this technology even more attractive than it would be if it were a common place occurence. Similarly, mastering and possessing the technology would seem to provide the user with power proportionally greater than would be the case in a developed country. Accordingly, when the gap to be spanned is so much greater, the potential for disaster is much greater, so that in some cases the high technology solution may actually be a step backwards. This is especially true when time proven non-computerized methods already exist. A non-computer example of this situation is the heavy investment in mechanized teak logging equipment in Burma. Machinery there causes more damage to the environment than would be the case with time proven logging methods using elephants. The mechanization may at first seem more efficient, but in the long term may be detrimental to the environment.

We already mentioned in the previous section how, especially with a customized computer system, a variety of support may be required to create an effective system that meets the users needs. In developing countries there exists a very siginificant additional problem: the lack of a support system for even the basic hardware and software, with a lack of hardware support as the most visible problem. Behind each system there must be a support structure for maintaining that system. When isolated it is certainly doomed to failure. A system must therefore be located within reach of a maintenance organization. Distanceavailability will be directly related to the systems performance and reliability. Options for a maintenance organization will be discussed in the next section.

The need for proper maintenance in developing countries is even more critical because of environmental conditions. One of the important factors in the man-made environment is the power supply. Power outages may occur frequently and abruptly. Therefore, every precaution must be taken to regulate the power, and safeguard the equipment against interruptions. Another factor in the man-made environment may be the quality of the telephone lines. Whereas in developed countries software diagnostics and maintenance is routinely performed from distant locations, this is probably not feasible in developing countries. The natural environment may pose other problems. Temperature and humidity conditions may far exceed normal operating conditions, creating a need for a special air-conditioned facility. The logistics associated with creating this facility may then become a stumbling block for a successful application. It is here that microcomputers may have a very important contribution to make, because they do not require a special environment and may even be ruggedized for field use under difficult conditions. An interesting account of

such an adoptation is found in an arcticle by Case (1984), who adapted a KAYPRO microcomputer for archaeological field use in Mexico.

With respect to developing countries one must further consider the absence of trained personnel. The software may have bugs and quirks, so that frequent consultations with the supplier are called for. If this cannot be done, than the operating personnel must absorb the problems through additional knowledge and skills. Users of hardware and software must be in the right frame of mind. They must possess a healthy degree of skepticism, and a realistic and cautious attitude as to what may be and can be accomplished with automated equipment. The old adage will always prevail: garbage in, garbage out, even though the garbage may look neat and colorful.

Finally a computerized system maybe operated successfully, but it may not fit its role in the infrastructure of the application. This pitfall falls in the category of putting new wine into old skins.

4 APPLICATIONS GUIDELINES FOR DEVELOPING COUNTRIES

The most basic and important guideline for computer use with respect to developing countries (and elsewhere) is to avoid the general pitfalls described in the previous chapter, that is, to recognize the unreasonable appeal of high technology, the aura of power associated with computing, and to avoid the leap into space. The main remedy is consumer education, and this is the reason why much of the report is devoted to a descriptions of systems, and the current state of technology as applied to forestry data processing. Education provides familiarity, and with familiarity the unreasonable appeal may disappear. On the other hand, computerized systems can be extremely fascinating, and to a degree this fascination may provide a healthy stimulus for progress in the application field. However, adequate knowledge and background will prevent unsubstantiated decisions. If the path to be taken has been well charted, then it can be travelled, one small section at a time. There will always be uncertainty, but at least it can be reduced as much as possible, and the risk factors can be evaluated. Covvey and McAlister have the following to say: 'the leap into space may effective sometimes in solving a major problem, but it does nothing to recover the time, effort and money that may be wasted if the change fails to produce the desired effect. A more rational approach to change as major as the introduction of automation, is to proceed in an orderly, stepwise, conscious and prepared fashion that lends itself to adjustments in direction as they become necessary.'

The guidelines for selecting the steps to be taken fall into the technical and management categories.

4.1 Technical Guidelines

4.1.1 Feasibility Study

When preparing for a computer application, one of the first and most important steps to be taken is to prepare a feasibility study. This study can result in a feasibility document, containing a recommendation for ways in which one can proceed or not proceed with the automation. The feasibility study must address the following topics at a minimum:

- o assessing project and user needs
- o possible hardware and software configurations
- o feasibility of meeting user needs with these configurations
- o results of potential system trials
- o how the automation fits in with the rest of the application

- o the economics of the application
- o human and software engineering aspects
- o privacy and security considerations

4.1.1.1 Assessing Project And User Needs

Assessing user needs is a very important step in the feasibility study. It is only by knowing what is needed and wanted than one can hope to create a successful system. Success results from recognizing a need and then satisfying this need with the right system. However in many cases user needs can be a 'chicken or egg' problem, because the user does not know the automated application, and may not be able to express what his desires are, or he may not want to commit himself. The answer to this problem is to involve the user as as much as possible in the entire development process, so that he can learn and provide input along the way. This is currently very much a trend in software design.

A useful exercise in assessing user needs maybe to trace through an actual manual application of a problem, and try to visualize how the same problem might be solved in the automated application. The author has attempted this with the example provided in Appendix A.

If user needs are very hard to pin down, as may be the case in a research situation, it may pay to consider other systems in other organizations or locations, that have identical objectives. One can then conduct user interviews and compare user needs in those organizations to one's own.

Sometimes it may be very hard to define user needs. because the need for the system is dictated by goals and objectives set by the people 'upstairs.'One may then be faced with the unpleasant task of designing the user as well as the system. However, in this situation one must seriously question these goals and objectives, if possible. There really must be a worthwhile cause for which a computerized solution is appropriate.

The outcome of the feasibility study should be a recommendation to: 1) scrap the project, 2) proceed with a non-computerized solution, 3) proceed with a computerized or partially computerized solution.

4.1.2 System Design

Once the decision has been made to proceed with the project, a system design is called for. It must address the same topics covered in the feasibility study, but in a more definitive manner. The term system in this context refers to the entire application, complete with its human interfaces, inputs, outputs, data sources etc.. Rather than just keeping the hardware and the software in mind, one should also be concerned with the additonal components that help make up the entire application.

4.1.2.1 Functional Specifications

The system design is arrived at through a system design procedure, and the first step to be taken is to translate the user needs and other concerns into a firm set of functional specifications.

For instance, when a Geographic Information System is involved, the functional specification may state how many maps of what complexity must be entered into the system on a daily or weekly or some other time basis. Or what kinds of queries are likely to be made of the system, and with what frequency they will be made. Some initial storage capacity computations maybe made, and the contents of the database may be assessed. If not already done in the feasibility study, then frequent consultations with suppliers and other users may be necessary at this point, in order to obtain a realistic view of what is possible and what is not possible.

The purpose of the functional specification is threefold: its serves to organize the requirements of the users, it forces one to put thoughts and dreams on paper in concrete realistics terms. Contigencies that might otherwise be overlooked must be considered. Secondly, functional specifications on paper can be presented to users and managers of the system, and people otherwise involved in its review. This brings gaps in one's thinking and other misconceptions to the surface, good work can be reinforced,and misleading activities can be exposed. Thirdly, functional requirements are extremely essential for requesting systems configurations and quotations for hardware and software from system developers and suppliers.

Following Covvey and McAlister, the topics that need to be adressed in a set of functional specifications can be assigned to eight general categories:

1. Introduction
2. The External Specification
3. Software Tools
4. Hardware
5. Applications Software
6. Documentation
7. Schedule
8. General Terms of Acquisition

The introduction must contain the goals and objectives for the system. The introduction may also contain a summary for the entire functional specifications. The external specification describes the external features of the system. By external features is meant the functioning of the system where the hardware and software is

perceived as a black box, and nothing is said about the way in which the functions are performed. This section will therefore contain a description of what the system is expected to do in its applications, along with all the required inputs and outputs. It describes the general range of capabilities. Minimum and maximum levels of performance can be specified, and general throughput requirements can be listed. It is also useful to anticipate and describe any future changes in processing capabilities, and to state the future expansion potential that the system must have.

The section on software tools should mention the types of system software that will be required. The type of operating system should be specified (single user, multiuser, real-time, etc.). If not known there should be enough detail for the supplier to make a suggestion. The types of programming languages and system utilities should be specified. Specialized systems such as database management systems, query languages, report generators also fall in this category.

The hardware section does not have to be detailed, because the composition of the hardware is what needs to be obtained. It should have good general characteristics, such as the number of terminals, needed disk capacity, tape units required, size of digitizing table, etc.

The applications software section will probably be at the heart of the specifications: this is where the requirements for all special applications software should be presented. It must contain the nitty gritty details for the work that the system is expected to perform, as for instance the format shape and size of the input data, and report and map formats for output products. The types of analyses that one expects to perform should be described: if not in the perform of a complete set of functions then in the form of user scenario's, and preferably a combination of both.

If special custom software is called for, the applications section may contain specifications for this software, and the way in which it must be written. For instance, one may specifiy that the software must be written in a high level language, and that structured programming techniques must be used.

The section on documentation must specify the documentation standards. User manuals are most important, but all other kinds of supporting documentation must also be requested. Detailed system documentation is required for programmers to make changes and debug the system.

The schedule section contains the schedule for getting the system operational. As far as the supplier is concerned this schedule should begin at the date on which the contract is signed. The proposed schedule must contain installation and delivery dates. It is wise to include a period of acceptance testing, and a final date for accepting or rejecting the system, along with a specified acceptance procedure. Most importantly, the schedule must contain a series of clearly defined landmarks, so that either progress is made, or the process can be terminated. Payments to suppliers can be tied to the delivery and acceptance portion of the schedule.

The arrangements that are acceptable for acquisition of the system are spelled

out in the terms of acquisition section of the functional requirements. A system may be leased, or it may be purchased outright. Maintenance may come from from the individual suppliers, or the system integrator may be responsible for maintenance. Special software may have to be developed on a fixed price basis or one may be willing to accept a cost plus fee arrangement.

Designing a system through functional specifications, with the selection of hardware and software, is a difficult task. Calkins (1982) states that 'the design problem faced by system designers is one of creating a functionally adequate system from a generally very loose set of specifications or requirements. The characteristics commonly found in the beginning of a design process are that there has been no definitive formulation of the need, there is no clear idea when an adequate design will be reached, there is no indication as to what the 'correct' solution will be, and finally there is little idea regarding how to test a design to see if it adequately meets the need.' The entire process is therefore sometimes better relegated to an agent or consultant, depending on the size of the project. Also, there are publications that deal with the design problem. For geographic information systems see for instance 'A Pragmatic Approach to Geographic Information System Design' by Calkins (1982), from which the above quote was taken. A consultant may charge fees equal to a significant fraction of the total cost of the initial purchase, but this may be well justified, considering the effort, cost and embarrassment saved.

Based on the functional specifications, one must select the proper hardware and software. The specifications can be translated into a request for proposal or RFP. Prospective suppliers can then suggest systems, configurations, maintenance arrangements, etc. The request for proposal with the specifications can be written by a lawyer, to serve as part of a future contract, thereby providing the buyer some legal protection. The RFP should deal with four major points: the specifications or buyers basic needs, the hardware to meet the needs, the software to meet the needs, and service and maintenance.

A hardware and software selection must then be made based on the response received from the various suppliers. The supplier will probably select the configuration that in his view is optimal for dealing with the functional specifications, but the consumer must scrutinize the responses to the best of his ability, to make sure that all factors have been considered, and in the case of multiple responses, to make a selection. The selected hardware-software can then be incorporated into the overall system design.

There are so many different kinds of computer hardware and there is such a diversity of systems software, that it may difficult to select an optimal hardware-software configuration. In the development situation, this problem may be somewhat less, because the requirement for support may drastically narrow the range of selection. There may be several competing turnkey systems that can satisfy the project requirements. The best system may then be selected through a careful comparison of the functional requirements and the systems stated capabilities. When doubt remains, it may be necessary to verify claims by performing on site testing, with data provided by the prospective user. For demonstrations that are done on a no-cost

basis examples must be small but representative. When the choice is finally narrowed to one or two systems, it may be necessary to run larger 'volume' tests to see whether the candidate system can meet the projected workload. Such experiments can be cast in the form of small pilot projects, paid for by the prospective buyer, or by the supplier, if the latter feels that his chances for an award are sufficiently large.

To make the purchase, the previous RFP with the functional specifications can be revised, and perhaps be tightened legally, so that it can form the basis of a contract. The obligations of each party, and the costs that have been mutually agreed upon are specified in the document, and it is signed as a binding agreement.

One word of caution is in order with respect to contract enforcement. Unfortunately a contract itself can guarantee very little. If one is forced to resort to the letter of the contract it is already too late. Progress can be made only in the spirit of mutual cooperation. Fights over words in the contract lead nowhere. The supplier must be aware of the general objectives and goals and have a feeling for the basic principles underlying the contractual specifications. The contract then serves as a rather precise focal point through which both parties understand each other, and through which they agree on the same ground rules. In the case of gross negligence or damage, a contract may provide some protection as an ultimate resort, but it generally does not pay to use it to enforce the fine points of the specification. It is therefore exceedingly important to check on the reputation of the supplier regarding the satisfaction of other customers. This may be even more important for developing countries, where the supplier is situated in a different country with a different legal system, far removed from the problem site.

4.1.2.2 Software Engineering

Although the development of the functional specifications and the selection of hardware and software are the most important functions in a system desgin there are other important technical aspects that one must be mindful of. These are: software engineering, human engineering, privacy and security, maintenance, and database design.

If a project requires custom software, either written by the supplier or the customer then one cannot help but become involved in software engineering. Good software is engineered, not haphazardly created. The subject is rather a complex topic, with many articles and books devoted to it. Software development is expensive and often may well be the most expensive component of a computer system. This is so, because it takes a long time to produce finished programs that work according to specifications. Almost two/thirds of the time may be spent on software maintenance, that is, coping with problems and errors during the life of the system.

There are four phases in software development: design, coding, testing, and maintenance. Major design techniques are: top down development, in which a system is defined starting at the top, or outside, and is subsequently refined to the greatest detail. Another technique begins at the bottom. In reality a combination of both methods will be used. Structured programming is another basic development

technique. GO TO's are avoided by using specialized control structures. This technique lends itself well for top down desgin. Other techniques used are HIPO (hierarchical input, process, output),and stepwise refinement.

The organization of a programming team is a subject for much debate. The Chief programmer team is one organization, in which there is one chief, one backup chief, a programming secretary, and a number of team programmers.

Documentation is extremely important throughout the life of the programming project. There should be three kinds of documentation: user documentation, program documentation, and system documentation. Contrary to popular thought, the coding phase actually takes the least amount of time in the programming project. Therefore, coding can go relatively fast. Considering all the other related efforts however, it has been estimated that an experienced programmer can only write 20 lines of fully tested code per day.

In the testing phase, the written code is tried out, and problems are eliminated as much as possible. Testing may occur in different phases, such as module testing, subsystem integration testing, system integration testing, volume testing, etc. Problems of all sizes and shapes may occur. Many will be in the form of 'software bugs'. But other types can also surface, such as lack of speed and adequate performance, space shortages, user unfriendliness, lack of recovery capability after system failures, etc. It is not merely a matter of detecting erroneous code in a program, because a program may work perfectly, but it is also a matter of seeing wether the program lives up to user specification. This is another reason for keeping the user involved in all phases of the project, including testing.

Testing will initially remove many problems, but many others may remain dormant, and must be removed over time. A system must therefore mature through constant used. This puts a burden on the user, who must cope with these errors by reporting them, and by finding ways around them. The problematic process of correcting errors after system delivery is called software maintenance. This is a very misleading word, because nothing is being maintained (Covvey and MacAlister). The word simply denotes the eventual detection and correction of errors that should not have occurred in the first place. In developing countries one may be very much isolated from the sources of software maintenance, so that it becomes even more important to use reliable systems that have withstood the test of time.

However, when a software maintenance program can be obtained, and the logistics problems can be overcome, one should try to implement a maintenance program at all cost. Covvey and MacAlister state: 'It is wishful thinking for the user of a computer system to avoid a software maintenance contract, in spite of high cost. The user can pay the developer now, or pay even more later.

Good project management is very important when developing software. For the user it is especially important to remain in close contact with the developer. The only practical way to assure steady progress is to have demonstrable milestones. Assurance that the system has been 'coded' or is 90% complete may not mean

anything. Coding is only a small part of the total effort, and most of the work must go into testing, debugging and software maintenance.

Considering the current state of software development, and the state of software in general, there are some unsettling truths that one must always keep in mind when considering automation. First, software cannot be totally proven correct, and so the product one buys is a priori faulty. One expects the product to function correctly, but there is absolutely no guarantee that this is so. Hence the importance of continued maintenance. Furthermore, until a program or system has a proven reliability, a high degree of suspicion is absolutely mandatory. It is highly advised to always verify outputs in several ways, and to apply consistency checks wherever possible. Secondly, good software development practices are the exception rather than the rule, one can therefore expect to find sloppy programming with its pitfalls. When dealing with geographic information systems, where the system must deal with two-dimensional data, these considerations carry even more weight, because the problems may be far more intractable than in other type systems. This is particularly so for vector-based systems. Vector based systems are usually far more error prone than raster based systems. Thirdly, one must realize that errors may propagate themselves, especially where database systems are involved. Once the database is in a bad state, it may be very difficult to recover.

As said before, the solution to all the above problems is to be properly educated, so that one can be properly skeptical, and play the 'devils advocate' as much as possible.

4.1.2.3 Human Engineering

One of the most important technical considerations concerns the way in which the computerized application fits in its environment. The evironment can be seen as the people that operate the system, the data inputs and outputs of the system, as well as the wider aspects of the project and department that is served by the system. The quality of this interface is what may make or brake the system: set it aside as a white elephant, or make it a widely accepted way for accomplishing project tasks. As humans must necessarily deal with the machines, the topic to be discussed is that of human engineering. Humans cannot be engineered, they must be educated and trained, but the system must be engineered to be as humanly acceptable as possible. With the current state of technology, there is a real problem in this area. Computers and humans are basically incompatible. Machines are impersonal, unfriendly and dumb. They carry out to the letter what they are instructed to do, and sometimes wreak havoc when a wrong turn is taken. Computers are one-dimensional machines, in which every step depends on the correct execution of the previous step. One mishap, and the consequences of the remaining millions of decisions may be entirely unpredictable. Therefore, one must take all kinds of precautions, such as is the case in properly engineered software. The computer does not relieve one from having to think. It may dispose of routine repetitive jobs, but otherwise requires the user to be more alert for problems and false results than would be the case otherwise.

A computer user forms a mental psychological model that, to him, explains the

requirements and responses of a computer system. This model is based on documentation, training, past experience and communication with other users. The success of the human machine interaction depends on the ease with which such a model can be formed, and the degree to which the machines actions deviate from the model. For an office automation system, one such model is that of a desk with documents, file folders, trashcan, etc.. This model is currently successfully used in various computers. It is expressly human engineered. Other systems are not directly based on a real world model, so that the user model may be rather abstract. In that case simplicity is all important. For instance, systems that require the maintenance of a large number of different files tend to be confusing, because it is hard to keep track of the reasons for file creation and deletion, and it may be hard to keep track of the file whereabouts, using a mental model. This is one reason for the utility of a hierarchical file directory.

Two areas of automation where human engineering is most important are those of input and output. Inputting data into a computer system is probably the most error prone and difficult area of interaction with an automated system. Software and hardware must both be geared to an acceptable input process. This is especially true in the area of geographic information systems, where the majority of the project funds may be used for map digitization. This is where the so called 'grunt' effort is made. All to often, the magnitude and the scope of this effort is downplayed, because the time estimates may be hard to believe. It is especially important at this stage to be sure that the data entered are 'clean', and not to postpone this scrutiny. Erroneous input can be a serious source of grief in the subsequent analysis stage. For instance, entered map data must have a clean topology. Field data from a forest inventory must be subjected to as many checks as one can possibly think of. One could almost speak of 'data engineering' in this regard, in analogy to software engineering. The human engineering aspect of a system bears on the potential for making errors with the system, for detecting them, and determines the ease with which they can be corrected. A badly engineered system may require the subsequent re-entry of an entire unit in which an error occurred, rather than allowing a localized update. Another aspect of human engineering in input is whether a system allows for a variety of users with different proficiencies. This aspect has been discussed in the section on raster based systems.

Human engineering in output is equally essential for the success of a system. For example, many image processing systems have no provision for hardcopy map output. Yet clearly, a colorful picture on a display terminal cannot be used for further field work. It is essential to consider how the outputs of one system are going to be used as inputs for the next, whether human analysis, or further processing.

Error reporting is a very important facet of human engineering. One should not have to put up with cryptic error codes. Even more important however, is some kind of error reporting that gives one a clue as to how to solve the problem. In the worst possible situation one receives no error report at all, when something has indeed gone wrong. One may then not know whether the machine is doing useful work or wasting the users time. Well thought out systems will provide progress reports, as the work proceeds. The user likes to be informed as to the current state of the process. Not

letting the user know is equivalent to waiting in a doctors office, not knowing whether one has been forgotten, or whether the doctor has been delayed due to emergency surgery.

The most important objective for human engineering should be to design a system that has a minimum potential for frustrating its users. A system has to be user friendly, but above all, it has to be user tolerant, and this means that it must be reliable. Users will tend to forgive a system that puts them to a great deal of inconvenience as long as the system is reliable. System reliability is often taken for granted; but no other feature of human engineering is so important as the reliability of a computer system. This holds true even more for computer systems in developing countries, where the degree of maintenance may be less due to the lack of a local servicing organization.

Unreliable systems waste a great deal of a users time. Projects that might otherwise take a small amount of time can become major obstacles in the face of unreliability. Subtle errors can have ripple effects, that are not immediately obvious, but may generate a need for considerable backtracking and rebuilding of results that had been already established. A thoughtfully engineered system will provide 'fail safe' routines through which most of the users work can be salvaged in case of failure, or which at least guarantee that a previously consistent state is recreated. However, even with this type of protetction it is still extremely important to execute adequate backup methods. One should always investigate the provisions for failures in any system. If they exist, then one can be assured that the system has been thoughtfully engineered, but if not, the user should be seriously warned. Computer systems are only machines, and they are guaranteed to fail at embarrassing moments.

Just as computers must be designed to be human oriented, humans must be adapted to the machines in a certain way; this means education and training. But there may also be some psychological resistance to overcome. There may be resentment, because of the radical changes that are brought about by automated methods. In the training phase, productivity may be low, and frustration high.

A special effort may have to be made to convince individuals of the personal benefits that may be obtained by using the computerized approach.

In summary, it can be said that human engineering is the most important aspect of the automated applications, because humans use the machinery, give it a good or bad reputation, and otherwise decide the fate of the application.

4.1.2.4 Privacy And Security

A further technical consideration already mentioned in several instances, is the issue of privacy and security. Privacy is a very important topic when storing personal or otherwise sensitive data. Security has two aspects: 1) to make the data secure against intrusion by unauthorized people, and 2) to guarantee the survival of the data in the face of threats such as human error, system failure, vandalism, and natural disasters. We will briefly discuss this last and more important topic.

We already mentioned the importance of fail-safe methods as well as the use of appropriate backup measures. Duplicate copies of all important data must be kept. A system can be backup totally, say once a week, and changed files can be written to tape every day. In designing backup procedures one must consider what the most amount of data or work is that one can afford to loose in the worst case situation. This loss must then be weighed against the cost of doing the backup activities with the required frequencies. One must consider what can happen to the backup media also, and so, for instance one must not store the backup data in the same room with the computer. Both the backup medium and the computer and on-line physical storage must be kept physically secure. For larger machines this means air conditioned rooms with fire protection, voltage regulators, emergency power generators, etc. For microcomputers such stringent precautions are perhaps not required, because the potential loss is only a fraction of that of a mini or mainframe. However, one must still protect the data by backing up the floppy or hard disks, and perhaps storing this backup data in a safe place such as a different building, safe, or even refrigerator (they tend to survive fires). The environment required for mini and larger computers can easily double the cost of the computer hardware itself. That is a very important reason for trying to fit the application on a microcomputer, if at all possible.

4.1.2.5 Maintenance

The importance of software maintenance has already been mentioned. Software maintenance is more or less synonymous with ongoing software development and improvement. Hardware, maintenance on the other hand, is real maintenance, and falls into two categories, preventive maintenance and repair. Suppliers usually have contracts for preventive maintenance and repair. With such a contract the machine may be guaranteed to be operable for a certain amount of time each month, and the response time in case of failure may be guaranteed. If such a contract can be obtained, it is certainly recommended. However, the cost of a maintenance contract may be high, and this is one of the hidden costs that must be considered when acquiring a computer system. In developing countries, it may be difficult to obtain a maintenance contract, and the user may have to be responsible for maintenance and repair. In such a case, one must send appropriate personnel to the contractors facility for training, and also maintain a supply of critical spare parts. One must keep in mind that the spare parts cost may be a sizable fraction of the total cost of the system, in some cases it may even be recommended to buy two sets of hardware, one to cannibalize to keep the other machine operable.

4.1.2.6 Database Design

So far, we have been concerned with the software and the hardware provided by suppliers. However, the system is not complete, especially in applications where database systems and geographic information systems are concerned without a data base; and hence one must consider the area of database design. There exists a large body of knowledge in this area where ordinary database systems are concerned, but little has been written about the optimal design of database for geographic information systems, because there are so few commercially available systems, compared to

regular database systems. Design considerations for commercially available database systems can be rather complex. We have touched on some considerations for forestry applications for an hierarchical and a relational database system (chapter 2). When designing a database for a relational database system, when the data must be maintained on an ongoing basis, there are certain norms to adhere to, in order to avoid trouble during updates (degrees of normalization).

For a geographic information system, these rules also hold, if the attribute database system is a commercial database system, but there are few standard guidelines otherwise. However, there is one very important rule of thumb, that should always be used when considering what map layers and attribute data to store in the system. There is an enormous tendency to store unnecessary data. In the section on input, we mentioned that a large part of the expense for the system, may be the initial loading with map data, and so a judicious selection of the data to be loaded is enormously important. The basic rule of thumb is as follows: begin with the question that will be asked of the system, and the problems that it must solve, and work the data requirements backwards from there. This seems simple and straightforward enough, but the author has seen many examples where many data layers were stored in a system, and only a fraction were actually needed.

Other considerations in the design of a database are what portions of the database should be on line, what parts belong in archival storage, and how to facilitate fast retrievals for parts of the data that are used frequently.

Geographic information systems usually revolve around 'layers' of map data. Sometimes it may be hard to decide what should be a separate layer, and what should be coded as attribute values.

A basic rule in this area, is to group data of like types, such as points, lines or polygons together in layers, and to make spatial data in these layers as mutually exclusive as possible, that is, to avoid overlap of units. Thus, for example roads and railroads may be stored in the same layer, but it is probably more advantageous, to separate them out. Likewise, if townships overlap with management circles than the two types of data must be stored in separate layers.

There are many other problems that one must consider when designing a database. The quality of the data is important. Once the data appears in computerized form, it may take on a credibility that is unwarranted. One must therefore avoid loading questionable data at all cost. Another problem that crops up when data from different sources are all loaded into the same system, are discrepancies in interpretation and registration. Quite often an important line that should be the same on a vegetation map and soils map, may be roughly the same, but there may be enough variation, to make simultaneous use impossible. The differences may then have to be straightened out beforehand, through a reinterpretation of the input maps.

One must also decide how to handle missing data, depending on whether the system can cope with missing data. Two types of missing data are recognized: relevant, but not available, or missing but irrelevant. For instance, one may have a

timber volume data item for each landuse type, but for certain types, such as water, this data is irrelevant.

When data ranges are categorized into classes, one must make sure that the thresholds occur at the desired junctures. For instance entering a slope class map with degrees of slope < 20%,20-50% and > 50%, does not make sense when the inquiries made of the system relate to < 10%,10-60%, and > 60% slope limits.

4.2 Management Guidelines

Technical and management problems are not easily separated because in the end, all problems have an effect on the success and the life of the project, and must therefore be considered management problems. In the previous section we discussed a number of technical problems with management aspects, such as issues related to procurement, contracts and maintenance, privacy and security.

In this section, we would like to mention three other management subjects, namely the economics of computing, managing the automated system and personnel training.

4.2.1 The Economics Of Computing

The economics of computing is a subject that is often neglected, or may be some token effort is made, but in many instances it is just not considered. However, it only makes sense to provide some economic justification for introducing computerized automation. Even when the benefits as a system may be intangible, such as for instance improved working conditions, more up-to-date information, etc., the costs will be very real, and can be evaluated beforehand. It will always pay to evaluate total costs, whether they can be compared to real cost savings, or can only be weighed against intangible benefits. The process of attempting to evaluate total costs in itself may prove a revealing exercise: in the process of systematically assessing all costs, many hidden costs may be discovered.

Two basic types of economic analyses can be made: 1) a cost-benefit analysis, in which real costs are measured against financial benefits, such as the elimination of personnel positions; 2) a cost effectiveness evaluation, where the effectiveness is based on both tangible and intangible benefits, which may even be negative, such as the loss of personnel employment for those people who are no longer needed.

But as the effectiveness side may be entirely subjective, the real benefit of an economic analysis may be the cost evaluation, which may turn out to be a shocking revelation.

In evaluating costs, one should try to think of all possible cost categories, which at a minimum should include the following: hardware, software, hardware maintenance, software maintenance, communication costs, supplies, environmental costs, miscellaneous costs, inflation, taxes, and the cost of establishing the database.

One should consider all the hidden costs of various arrangements, even when the system is leased, rather than bought outright. If maintenance is charged for on an as needed basis, then one should have some idea of the frequencies of breakdowns and the need for maintenance.

One hidden cost that usually goes unstated is the cost for the proper environment of the hardware. This may be an especially significant cost in developing countries, where a suitable environment may have to be created from scratch. This involves special electrical work, air-conditioning, voltage regulation, fire alarm, fire extinguishing system, special locks, etc. This is why a small microcomputer may have serious advantages in developing countries.

In the miscellaneous category, one may put costs for lawyers fees, travel to supplier sites, shipping, insurance, and many other operating expenses.

When the application involves a DBMS, or a GIS, then one important cost to be considered is the cost of loading the data to establish the database. Many man-hours may have to be expended digitizing map data, or typing in records. This initial expense may be the greatest stumbling block in getting the automated operation under way.

When considering possible savings in a cost-benefit analysis, one must look for all possible savings, and some may not be obvious at first. Other agencies or firms may be inclined to rent time available on the new system, and this may be a source of income. Or expensive supplies, hardware and furniture (such as filing cabinets) may not be required any longer.

In summary, it can probably be said that final evaluation for obtaining a computer system is probably not based on a favorable cost-benefit ratio, but that if a good ratio can be computed, it may certainly help ease the decision to buy.

Covvey and McAlister sum the economics issue up as follows: 'accurate forecasting of total costs and appreciation of the many ways in which a computing system can save money, will help plan for systems that will be used and this will go a long way toward guaranteeing the long-term survivability and success of the project.'

4.2.2 Managing The System

Once a system is in use, system development and maintenance is a never ending task. To cope with the ongoing work, at least for larger machines, there must be some kind of management structure. People will have to run and feed the computer system, and these people must be properly trained (see the training section). There may have to be an in-house computer team to maintain the system, in addition to the regular system users. Programmer teams may be needed, and perhaps even a high level manager specially dedicated to the automated application. The advantage to having a high level manager maybe that one has a direct link to the top. On the other hand, this can become a power position, through which the goals of the organization can be gradually perverted. A shift from real accomplishments to a preoccupation with data processing itself is one of the symptoms of conspicuous computing.

The automated system must be integrated into the total application, and this may necessitate changes in previously existing organizational structures. One issue that always rears its head is the one of centralized versus de-centralized computing. In support for a central approach one can present the following arguments. Theoretically it is easier to coordinate the computing needs of an entire organization through one central focal point. Central control facilitates the review and assessment of a variety of projects, so that priorities can be assigned. There may be economies of scale, because resources can be shared by different departments and organizations. A central facility will also attract more qualified and competent personnel. With a number of small applications, it would seem to make sense that they all share the same computing equipment. It may be the only way in which one can afford to do the projects. With centralization there can be more control, thereby preventing anyone project from squandering the computing resources. Projects that have no hope of success can be reviewed and terminated. Projects that are successful but have come to an end can also be terminated.

Notwithstanding all these advantages, there are some serious drawbacks to the centralized approach that outweigh them, especially in the development situation. As Covvey and McAlister point out, a single computer center in any institution can easily become the focus of a power struggle among individuals or departments because of the aura of power associated with computing. If many interests must be served by one installation, it is almost impossible to serve them all with the same fair objective goals. Some users will feel neglected or may be purposely misled with regard to the feasibility of their project or their slice of the computing pie. The classic forestry example of this situation is the company where resource management and corporate accounting must share the same computer. Resource management depends much on interactive scientific computing and number crunching, while accounting may be batch oriented with much I/O. These applications are so different, that either one or the other will be badly served. The resource operation is probably very important to the basic well-being of the company, but probably badly understood by the accountants, who are in control of the computer system.

In the development situation, a central computer system may seem like a good idea, but because of the different organizations involved may create such a bureaucratic hassle that the project may be doomed from the very beginning. Here, instead of an economy of scale, one may have a burden of scale. Because of the many organizations who may impose a workload on the machine, the computer must be larger. It therefore is in need of a special environment, will need approval from higher authorities, may require special import approval, etc..

All of these problems can be avoided by selecting a number of microsystems to handle individual applications. A times,it may even be possible to import micro's under the label of measuring equipment, thereby avoiding the specials problems associated with the word computer.

Current trends in system development certainly seem to be headed in the direction of smaller computer system connected through local networks, or attached to

a mini or mainframe, or computing service as the need may be. The risks associated with the micro system are also much smaller than those attached to a mini or mainframe machine. On the other hand if there are existing and well-functioning computers available locally, then the minimum risk strategy may be to use exisiting potential, until the application has proven itself, and the necessary experience has been gained. In the final analysis good judgement and common sense must dictate what will be the optimum mix of centralized and decentralized computing.

It may be difficult to adapt an organization to the necessities of productive data processing. However, some practical measures can be taken to insure that computers will be a productive asset. It is very important to have some formal management structure with well trained professionals in positions of authority. The existing management structure must be adapted to computer related activities. Also, there should be some kind of long term strategy, because conditions change so fast that a specific solution may never fully catch up with changing problems.

In summary it can be said that unless there is adequate planning and management, and unless there is a special manager in charge of the project, the automated application may fail.

4.2.3 Training

Well trained personnel is absolutely essential to the well-being and success of the computerized application. This is especially true with respect to developing countries where the support infrastructure may be poor, so that one must rely on locally available personnel. The well trained person will be able to anticipate problems and prevent them from happening, and will otherwise be able to help solve them when they do occur.

A computerized system should be robust and user tolerant, but there is a limit to the protection that can be built into the system itself, so that for the remainder one must rely on a well trained user. A highly trained user will also be able to exploit a system for everything it is worth, and thereby may be able to provide significant savings in time and energy, taking shortcuts that he may only know about.

A successsfully operated system consists of two components, the hardware-software environment on one hand, and well-trained users on the other hand. It simply does not make sense to spend a small fortune on one, but neglect the other. Therefore, one should select the best potential users and provide them with the best training. If this training happens to be provided by the supplier, and the supplier is located abroad, then one must not let travel expense interfere with the training, rather such costs should be part of the original budgeting consideration when acquiring a system.

In selecting training, one must be careful to select the right kind. A person charged with writing applications software, does not need to know all the ins and outs of the hardware, although some knowledge probably will be useful. Training can be obtained at different levels, and the appropriate level should be applied. The spectrum

may range from basic user training on a system to trade and technical schools to university education. People who are properly trained also acquire a broader perspective, and thereby can apply better judgement in selecting new methods and identifying available options. This will prevent the user from 'reinventing the wheel', when this is not beneficial.

The phrase 'reinventing the wheel' is otherwise a cliche. At times duplication of effort should be avoided. On the other hand, re-invention may be in itself a valid learning experience, because after all, there is not better way to experience the principles and implementation of an idea than by doing the work oneself. In doing so however, one should have the understanding that there may be other better wheels in existence, and the well trained person should be open minded enough to take advantage of such opportunites as they occur.

Learning is very much an on-going process, and one should therefore stay in touch with other organizations and systems as much as possible, through professional literature, attendence of conferences, user groups. etc.. A list of recommended computer journals has been provided in section 7.0.

5 CASE DISCUSSIONS

5.1 Brazil MA-IBDF

The 'Ministerio de Agricultura, Instituto Brasileiro de Desenvolvimento Forestal: MA-IBDF' has a very active forest monitoring program operating in its economics department. Its objective is to assess the general situation as well as changes occurring in Brazil's forest cover.

The program has four major fields of responsibilty: 1) deforestation, 2) reforestation, 3) national parks, and 4) forest inventory. The program is most concerned with the 'legal Amazon' (Acre, Rondonia, Amazonia, Mato Grosso, Goias, Mato Grosso, Maranhoa, Para), but also operates in the southern part of the country (Rio Grande de Sul).

A major task that is being carried out by the program is to produce 1:100,000 and 1:250,000-scale forest cover maps which are widely distributed within Brazil. These maps are in blue-print form. They are produced by manual interpretation of Landsat maps and aerial photographs. The project mostly uses black and white images of Landsat bands 5 and 7, but occasionally radar maps produced by project Radambrasil are interpreted.

Recently, the program has started to use Thematic Mapper Images extensively, in the scales of 1:100,000 and 1:250,000, both in black and white and in false color, and this has caused a dramatic improvement in the quality of the visual interpretation.

Major equipment used by the project is a Spectral Data color composite projection machine (used only on an experimental basis), a Keufel and Esser KARGL reflecting projector, a Japanese LI-COR Model 3100 area measuring machine, and blue-printing equipment.

The project produces a great number of maps, which are used for a variety of purposes. Some examples of map use are:

- to provide information for the collection of taxes.
- to provide information for the establishment of national parks.
- to determine the area of new plantations in southern Brazil as a basis for government reforestation subsidy.
- to locate resources for charcoal manufacture.
- to monitor colonization patterns.

- o to establish drainage patterns and forest cover for planning road locations.

- o to provide an assessment of changes and human activity patterns in national parks.

- o to determine timber volume loss due to deforestation in the Amazon area.

Base maps are obtained from other sources, such as the 'Directorio Servicio Geographico' and Project RADAMBRASIL. The objective is to obtain repeated coverage. The state of Rondonia has been the first area where repetitive coverage has been obtained on a two year basis. High priority is given to very critical areas.

The program has been working in close cooperation with project Radambrasil in the use of its Intergraph system, to produce maps and reports of the entire Northeast. These maps have been recently published, and now being plotted by computer.

Comments. As stated by the program manager 'the work being done may be 10-15 years behind what is being done in other countries for the same kinds of objectives, but it is extremely essential.' Indeed, the project could probably benefit by the use of some modern digital analysis system, especially for some of the more research oriented aspects of the project's main objectives. Currently, the program is using Brazilian universities to help out with some projects. However, whether such equipment would be cost justified is questionable. The current operation seems remarkably effective and efficient. A major benefit would be a capability to perform automated classification for the purpose of assisting manual interpretation. Such support could be valuable in those cases where classes may be hard to distinguish on black and white imagery, and might yield a more uniform interpretation. However, deforestation patterns may be easily discernible, and this type of interpretation may therefore just as easily be accomplished with the traditional methods. Major drawbacks of a digital image interpretation system are the need for digital imagery, and the expense of an output device such as an electrostatic plotter for translating the displays into hardcopy output.

A more direct benefit could probably be obtained through the use of a microcomputer, to keep track of maps, images, orders, deliveries etc.. Such a system might be operated using a microcomputer database management system.

The commercialization department of the IBDF economics division has purchased a Hewlett Packard 9845B tabletop microcomputer. They are in the process of buying a small plotter and printer. A similar machine could be used by the forest monitoring project. Brazil has severe import restrictions for foreign microcomputers to protect its own computer industry, which has developed micros such as the POLYMAX, but apparently HP equipment can be imported. FAO/UNDP provided the funds for the HP9845B. Currently this system is used for mailing lists, and for keeping track of wood-product marketing information. The mapping project should be able to use a similar machine for similar bookkeeping purposes, and could probably use a small plotter and digitizer as well. Such a system could then be used for plotting map

coverages, and other simple spatial overviews or summaries. It would also be useful for performing the statistical analysis required for some of the special project work.

5.2 Brazil Radambrasil

Project Radambrasil began its operations in 1971, with as its main objective to obtain radar coverage and semi-controlled radar mosaics of the Brazilian Amazon region. The scope of the project has since been enlarged to encompass all of Brazil, and radar imagery for the entire country has been produced. The project has had a profound influence on Brazilian mapping and cartography. The 1:250,000 semi-controlled mosaics have been interpreted at a 1:1,000,000 scale according to seven basic themes: geology, geomorphology. soils, vegetation, capacity of natural resources, metalogenetics, potential of hydrological resources. For each mapsheet area a volume containing ancillary information has been compiled. Today, most of the work is finished, 27 volumes have been published, the remainder are in preparation for publication. Each of the volumes contains invaluable resource information, where none had been available before.

Several years ago, because of a concern to make all the information available in digital form, project SIGA (Systema de Informacoes Geographico de Amazona) was initiated. The project was stationed in Rio de Janeiro. Its aim was to create and store a raster based database for the entire Amazon region. This project initially used IBM equipment and line printer output. In 1981 a decision was made to scrap project SIGA, and to move the information division to Radambrasil headquarters in Salvador, Bahia. A VAX 11/780 Intergraph system was purchased, complete with six workstations (including two high performance stations with extra memory), one plotter, several lineprinters, various alphanumeric terminals, an array processor and a graphics processor. By late 1983 the system was in place, and a staff was experimenting with the system, trying to master its various components, and designing a database for the Radambrasil data. Production digitizing (possibly in 24 hour shifts was not anticipated until 1984).

The initial objective for the system is to act as a digital 'warehouse' for the thematic maps of the project. The map layers to be stored in the system are organized around seven themes: 1) cartography, 2) geology, 3) geomorphology, 4) vegetation, 5) climate, 6) soils, 7) aerial photographic coverage. Each theme may ave as many as 14 separate layers. There will be a maximum of 63 layers per theme.

Comments. The Radambrasil Intergraph installation is one of the most complete and modern of its kind in South America. The total cost for the hardware and software was $708,000, a considerable investment. This probably does not include other costs such as site preparation and miscellaneous expenses.

A few observations can be made with regard to the procurement and operation of a system of this magnitude. The first is to point out the need for adequate training of personnel. This is an absolute necessity when one must operate a large and complex system where the cost of entering the data can easily exceed the cost of hardware and software. Software is never free of bugs, and will therefore require continuing

maintenance. In this context it is very important to know where the quirks and bugs are so that unexpected erroneous results are not blamed on user error. This knowledge is best obtained through training provided by the supplier. It is probably not cost effective, even when foreign travel is involved, to attempt discovery of the functionality of the system through trial and error.

The second observation is that an appreciation for the complexity of handling spatial data must be gained,in order to realistically assess what can be accomplished. The Intergraph is essentially a CAD/CAM (computer automated drafting/computer automated mapping system). Such systems are amazingly effective for the digitization and display of line work. However, in order to manipulate map data in a GIS mode and added dimension comes into play, namely the map topology. For instance, to calculate the area of a region, one must know the outside of the region (outer polygon), as well as the outside of the regions situated inside the outer polygon. This relationship is not present in the linework, but must be discovered through special spatial processing. Intergraph uses a special software package for this purpose called GPPU (General Polygon Processing Utility). The use of this package requires another level of processing, for which personnel must be trained, and the complexities of which must be understood to gain an appreciation of what GIS functions can be realistically accomplished with a CAD/CAM system. A problem in this regard is that adequate software support for the CAD/CAM functions maybe available at a national level (Sao Paulo), but that it may be difficult to get the same support for the more specialized functions such as GPPU, simply because they do not have the same usage as the CAD/CAM functions. This is another reason for getting initial training directly at the suppliers headquarters, even if international travel is involved. This will then also allow for setting up the required contacts for further software support.

The third observation relates to knowing the purpose for which the system is to be used, once data are available on the system. Currently, the Intergraph is only designated to serve as a digital storage depot. Not knowing what questions the system must be able to answer may cause unnecessary data to be stored, and probably also creates uncertainty as to the priority with which data layers must be entered.

Summarizing, it can be said that the Radambrasil Intergraph system is indeed a very modern system that may have an enormous potential for solving the many serious problems associated with the development of the Amazon region and other Brazilian development regions such as the drought stricken northeast.

5.3 Burma Forest Inventory

Burma has a very long tradition of forest management. The Burma Forest Department, founded in 1856, is the second oldest tropical forestry department in the world. Management plans have been in existence for a long time and are both voluminous and rigid. In recent years however, all previously private timber concessions have been nationalized, and a State Timber Corporation performs all logging and extraction. Instead of the cohesive Forest Department, there are now three separate entities: the Forest Department, the State Timber Corporation and the National Parks Department. All three are involved in management planning. A

reorganization of the countries management units has presented another difficulty in carrying on the long established tradition of quality forest management. For forest management purposes, the country had originally been divided into states, which consisted of divisions, which in turn were divided into management units, based mostly on a division in natural watersheds, following natural lines such as rivers, channels and ridges. Now the states are divided into townships (314 in all), which may follow arbitrary boundaries. This type of boundary creates a management problem, because logging operations still will follow the natural boundaries, but the reporting units will not.

Within this context a project has been established called the 'National Forest Survey and Inventory', with an objective to provide comprehensive forest resource information (inventory and maps) on a country-wide basis, and to develop this inventory information into a computerized information system that would be kept up-to-date through additional monitoring and recording of logging activities. In the ultimate system there would be three levels of information. At the first level there would be 1:250,000-scale state wide information. At the second level there can be inventory and pre-inventory data at the 1:50,000-scale level. The pre-inventory data result from a reconnaissance type inventory. Mean volumes can be based on end-use classes and end-use types, broken down by townships. The third level is at the working plan level, at which there can be an enumeration. The idea would be to provide up-to-date information to the system at the 1:250,000 level through satellite imagery, and to further update the volume data with growth function, and through incorporation of records of the Timber Corporation.

To support the project a DEC VAX 11/750 is being acquired. This machine will be used for processing current inventory data, and for establishing the above information system, as well as for keeping track of the Timber Corporation information such as the status of logging machinery and animals. The VAX is funded through FAO/UNDP. The computer will be housed in a completely new building with a specially designed computer facility. The total outlay for the machine will be over $1,000,000, invested in the computer, spare parts, and building.

Comments. Many of the conditions discussed in chapters 3 and 4 can be observed in Burma. The computer has an aura of power, enhanced through several factors related to the development situation. Burma has an extreme shortage of foreign currency, and as of the fall of 1983, had only one other VAX. Computer purchase decisions are made at the cabinet level.

A machine like a VAX needs adequate support structure and this area also turns out to be problematic. The distributorship is Indian, and the Indians have problems in obtaining spare parts. Dealing with DEC directly has proven difficult, and they are not very responsive.

The intended uses for the VAX are many, but one of the first real applications will be to process the forest inventory data. From this point of view, considering the power of the VAX, and the effort required to install this machine, one might regard the installation effort as distortion and a preoccupation with computing, which takes

valuable time away from the forest inventory. This distortion not only relates to time and effort, but also extends to the enormous capital outlay for the machine and its building, while for instance the inventory related mapping program proceeds on a shoestring budget.

A microsolution could have been found, at the start of the inventory project, but by then a commitment to the VAX had already been made. The import advantage, and avoiding the hassle of a special environment, are the major advantages of the micro system. For instance, a Micro PDP-11 can accomodate a hard disk with a capacity of 31 Megabytes, and this may be adequate for forest data processing.

Plans for the other VAX applications seemed vague with, as yet, little thought given to the software and expertise required for doing the job. It would seem that a feasibility study is needed, to be followed by a system design as mentioned in the Technical Guidelines Chapter. The Forest Department is currently understaffed and underequipped, and this may pose a problem for getting adequately trained people to develop the applications. One Burmese idea is to develop an entire information system from scratch. There seems to be little realization of the enormous software and engineering effort required to accomplish such a feat.

When the VAX will be installed, and with proper maintenance, Burmese forestry will have a very fine computing facility at its disposal. Then many new applications can be implemented without any worry about hardware and maintenance for those testing the application.

5.4 India FSI

The Forest Survey of India is located in Dehra Dun in the State of Uttar Pradesh, near the foothills of the Himalaya Mountain Range. It has only been in existence for a few years, and was formerly known as the Pre-Investment Survey of India (P.I.S.F.R.). FSI is charged with the on-going monitoring of India's forest resources on a national basis. The objective of the Survey is to know the growing stock on a national and state level, with 10 percent accuracy at the 95 percent probability level. To achieve this goal the Forest Survey is conducting a national forest inventory, with a systematic sampling design based on rectangular cells. Two plots are located in each cell, of 1 ha each. The first plot is selected randomly, the second is opposite the first, at the same distance from the center of the cell. The current plot density is sufficient to allow for make area estimates. Monitoring the entire country is estimated to take twenty years with the current methodology, so that areas are prioritized based on prior knowledge of the forest resources. Stratification through aerial photography or other remotely sensed imagery is currently not used, because aerial photographs are not readily available when needed.

The Forest Survey also has an intensive mapping program. Maps at a scale of 1:50,000 are interpreted from aerial photographs, but the Survey is also trying to develop a 1:1,000,000-scale map for the entire country with a minimum mapping area of 40 square kilometers. The map will be based on Landsat imagery. Special studies and research are performed on smaller areas, resulting in trial maps at various

scales.

Data processing for the forest inventory has been carried out with a number of different computers at different locations, as FSI currently does not have a computer system of its own. It has some old IBM verifying, sorting and collating equipment. FSI began its work by sending its first data sets to the Royal Forestry College in Sweden for processing, but within a year developed its own data processing system based on the locally available IBM-1620 computer of the Planning Commission and IARS, Pusa, in addition to Honeywell H-400 and IBM system-370 of Oil and Natural Gas Commission (O.N.G.C.).

Recently the Forest Research Institute, also located in Dehra Dun, acquired its own computer, a System-332, built by the Electronics Corporation of India Limited, modeled after the French IRIS system, of the French Computer Corporation, which has since been absorbed by Honeywell-Bull. Ironically, the IRIS was modelled after the IBM 370. The System-332 is 32 bit machine, with 256 Kilobytes of memory, 2 tape drives, 2 disk drives, 1 card reader, 1 line printer, and 1 tape to floppy disk converter. There is no plotting equipment. There is a proposal to install local terminals at the Survey. However, as the System-332 is based on older technology (the older models still have core memory), it is not clear whether multiple users can be accomodated by this machine. As the System-332 is of Indian make, there is not much existing software that can readily be used. However, there is a linear programming package, a PERT package, and a hierarchical DBMS system, that are currently installed on the computer. The System-332 has rather poor documentation and poor diagonstics. FSI plans to use the facility for future processing of inventory data.

Comments. The choice of System-332 computer system by the Forest Research Institute and its use by FSI presents a dilemma typical of several large developing countries, in particular India and Brazil, where one can either choose to support the national computer industry, and often buy obsolete technology, or try to obtain modern foreign equipment. Quite often the latter alternative is not a choice, as severe import restrictions limit the use of computer equipment. If a choice is possible, the advantages and disadvantages must be carefully weighed before a selection is made. On the positive side are the support of a national industry with its accompanying employment and a source for national know-how and technology development, as well as the needed local support structure to keep the machine operational. On the negative side is a sacrifice of a great deal of modern functionality available with the latest technology.

Undoubtedly, the weights to be attached to the advantages and disadvantages should be determined by the importance of the project's objectives. Whether the Forest Survey was in favor of procurring a System-332 is not known, but in any case, current developments in microcomputer make the restrictions resulting from the decision to buy a central mainframe computing system not as severe as they might once have seemed. Because of the microsystems low cost there is currently more of a potential for local distributed computing. In the case of FSI local micros could probably be profitably used for initial data editing, so that the obsolete IBM equipment could be retired. A major advantage would be the opportunity to interactively edit and check the

incoming data. Data could be stored on floppy disks, which could then be transferred to the System-332 through its floppy disk reader. Final processing could be performed on this machine, and data could be made accessbile through the Hierarchical DBMS system. There are a number of microsystems available in India, all nationally manufactured and designed around imported chips. For instance, UPTRON Produces the S800 and S850 microcomputer. this machine has 128 Kilobytes of main memory as well as two floppy disk drives. Cost of the S850 is approximately $10,000. The Indian Institute of Remote Sensing uses an UPTRON S800 to do planimetric block adjustments of 9x4 planimetric blocks. However, as pointed out in the technical guidelines chapter such a move should not be made merely for the sake of using improved computer technology, the pros and cons should be seriously considered. Microcomputers could also be used for other purposes, such as facilitating the mapping research and development at FSI. Apparently it is very difficult ot import foreign microcomputers into India, even gifts are not acceptable.

5.5 India Wildlife Institute

The India Wildlife Institute also has its headquarters in Dehra Dun, India. The institute currently has no computerized equipment, but has two objectives for which a computerized approach may be appropriate: 1) U.S. Fish and Wildlife Serivce type habitat evaluation (HEP), and 2) the development of a database for wildlife research information. An image processing GIS type system could be used for the first objective and part of the hardware for this system could perhaps also be used to accompolish the second objective.

The director of the Institute has proposed the use of a RIPS system, because he has seen this system operate in the U.S. RIPS is a small microcomputer based image processing system with a number of GIS functions developed around a Z-80 microprocessor. The system has 256 by 240 display unit and can hold the equivalent of 20 7.5-minute quadrangle maps at Landsat resolution on one dual density 8' floppy disk. The system has been designed as a workstation, to be connected to a larger computer, and the input of maps and image data to the system is one of the problem areas. Input must necessarily occur through another host system, with a facility for transferring maps to floppy disks at the required scale and with the required registration. The EROS Data Center can provide Landsat data on 8' floppy disks.

Other equipment that may be useful to the Institute is located at the Indian Institute of remote sensing (IIRS), also in Dehra Dun. This institute has recently acquired a Plessey MIDAS system (Multi-spectral Interactive Data Analysis System). It is built around a DEC PDP 11/23 LSI system, and has been developed in the United Kingdom. Peripherals for this system are: a Calcomp plotter, a Plessey dot-matrix line printer, a floppy disk unit, a magnetic tape drive, two 200 Megabyte disk drives, 1 color monitor, and 1 video terminal. The following equipment is on order: a Summagraphics digitizing board, a Hewlett-Packard 7585-B Plotter, a VT 105 terminal, a Tektronix 1415 B graphics terminal, a color graphics copier, and a Tektronix 4691. When this equipment will be installed, IIRS will have quite a complete set of hardware, capable of both input and output (digitizer board, Tektronix 4691, color copier). The MIDAS system as such is strictly a digital image interpretation system, with as its sole purpose the

classsification of Landsat images. However, another cell based information system has recently been installed on the MIDAS PDP 11/23. This is the USEMAP system, developed at ITC (International Training Center for Aerial Survey), Enschede, the Netherlands. This system is of particular interest for habitat evaluation, because it has several GIS functions, that although primitive, may be useful for this purpose. The system was originally developed for urban analysis, and is currently limited in its resource applications, but a new version is being developed at the ITC, and will be installed in Dehra Dun shortly.

Comments. To evaluate the Institutes options and wishes as short proposal (perhaps a pre-feasibility plan) was prepared, in which the following options were discussed.

Option 1. Under the first option, the Wildlife Institute would rely entirely on the facilities of IIRS. It would acquaint itself thoroughly with the USEMAP system, as well as with the Landsat MIDAS image processing capabilities. It could then begin to process new mapped data into the system through the USEMAP digitizing capability, and its line to raster conversion capabilities.

The advantages of this option are mainly threefold. First, the Institute would be free of any direct concern for hardware and hardware maintenance. Second, it would only be paying for actual use of the system, rather than having a permanent investment in expensive hardware. Third, it would be using a system with a hopefully well established support intra-structure. A major disadvantange for this option is that use and frequency of use, as well as future development of conjunctional capabilities would not be under control of the Institute.

Option 2. Under the second option, the Institute would first proceed as described, but when familiar with the IIRS system, would then acquire its own RIPS system to do local image processing at the Institute. Under this scenario RIPS is the terminal system, while MIDAS/USEMAP would be the host system. The host system would be used for input and registration of maps, for more time consuming analysis, and perhaps also for hardcopy output. The main problem to be solved for this arrangment is how to output MIDAS data into an 8 inch floppy disk format. The MIDAS PDP outputs data in a RSX-11M format, and RIPS can read a CP/M format. The problem was researched, and a potential solution was located in the form of a general reformatting program REFORMATTER available from MicroTech, a U.S. firm, at an approximate cost of $350.

A major advantage of the second option is the availability of the RIPS processing capability at the Institute. The total area that can be displayed bu a RIPS unit is 240x256, rather small but perhaps adequate for wildlife applications. However a powerful number of operators can be applied to RIPS images because of the versatile software.

The disadvantages of this option are several. Potential problem areas are: 1) cost of the RIPS unit, 2) support structure for the RIPS unit, 3) interfacing MIDAS and RIPS, 4) availability of trained personnel. A solution can probably be found in each of these

problem areas, but to obtain a useful system, they must be structured so that a step at a time is taken, to eliminate possible risks. For instance, it might happen that a RIPS is procured, but the interface with the MIDAS system cannot be worked out for some reason or other.

One careful way in which to proceed would be for somebody to thoroughly familiarize himself with the MIDAS system. This person could then be trained in RIPS use in the U.S., both at the suppliers location, and perhaps with another RIPS user, such as the U.S. Fish and Wildlife Service. In this training period a support structure can be built with both user and supplier support. While in training, this user could test the MIDAS RIPS interface. If the arrangements proved satisfactory, one could then go ahead with the procurement of a RIPS unit.

Option 3. Under this option the Institute would also proceed as in the first option, but would then acquire its own independent image processing system. This system would be self contained, and have its own digitizing and hardcopy output functions. It would therefore fall in a much higher price category ($50,000). Advantages of this approach lie in the in-house image processing, and the absence of a mix of systems. Systems with mixed components usually have a higher probability for problems and failures. A major disadvantage is the higher cost and the possibility that the system cannot be maintained in good operating condition

The main recommendation to be made to the Institute for its development of a digital image processing capability is to follow a thorough stepwise path in both understanding and use of a habitat evaluation system. Under all circumstances, the first option is recommended as a first phase in the development. Option 2 appears attractive because of the relatively low cost and the capabilities of the RIPS system, and because an already existing cooperative agreement with the U.S. Fish and Wildlife Service, through which financial assistence and expert training may be obtained. However, whatever option is selected one must be able to see one's way clear at each subsequent step, while keeping the associated costs and risks in mind.

5.6 India Forest Department

For purposes of forest management India is divided into states and territorial divisions. Each state has a conservator; the entire department is headed by a chief conservator. States are divided into districts which consist of working circles.

Folllowing the lines of this hierarchical organization, there exists a reporting procedure in which all information storage and transfer is accomplished with forms and ledgers. Some reporting steps may take several years. Some of the forms in use were devised by the British, and have not been changed since. The reported data have a tendency to get locked up at the State level, and so there exists a severe lack of feedback with regard to national planning decisions.

Throughout the reporting process there is much detailed information. This is espcially apparent at the working circle level. Working plans are thick volumes, with many different types of information about the forest and its environment, ranging from

botanical information to demographical and sociological data. The initial composition of a working plan may take many years. The plan consists of three parts. Part I contains general information (geology, soils, forest, climate,etc.), Part II contains prescriptions, and part III consists of Appendices (mostly data collected during field work). Every ten years, a re-enumeration of the forest is carried out. The working circle is divided into compartments, and for each there is a description of the boundaries, area, soil, situation, aspect, and a full accounting of its growing stock. Each compartment has five associated forms. Form No.28B has the general description, 28B(1) has a full compartment enumeration, 28B(2) lists the trees marked for felling by diameter class. Form 28B(3) is the compartment outturn form and 28B(4) has a full compartment history.

Comments. Computer systems at the state level would be very desirable. The forestry sector is behind other sectors in India with respect to computerization. However, the goal should not be computerization for the sake of computerization. Obviously vast improvements in data throughput should be possible, thus reducing the 'information float', with as a result a dramatic increase in the rate at which change can be brought about. Planning depends on good and timely data. For example, fifty percent of the domestic energy comsumption needs are met through fuelwood. If statistics were available to planners, indicating where and how it is used, planning could proceed with the allocation of new fuelwood plantations.

Apparently some thought has been given to the collection of computerized data from the states. The plan is to do this activity on the System-332 of the Forest Research Institute at Dehra Dun. However financial and technical assistance are needed. New forms must be designed and the data must be collected.

This would obviously be a step in the right direction. Computerization should first occur in the entire reporting process, where it best fits in terms of removing the largest bottleneck. Care has to be exercised to make sure that the automation would fit in with the rest of the non-computerized process, and that maximum benefit is obtained from having the data in computerized form. The data would probably need to be stored and accessed with a database management system. Whether the System-332 DBMS is up to handling this information would have to be a point of research. A feasibility study should be prepared.

Another point at which some automation might be introduced would be at the working plan circle level. For instance, a microcomputer could be installed for each working circle. All compartment data could be loaded (form 28B) into such a machine, using a database package. This would allow for a rapid review and updating of all pertinent compartment data. Such an application would certainly introduce computer technology at the groundlevel. As observed by Naisbitt (1984), most trends start at the ground level, while fads originate at the top. Initially such a project might not be economically worthwhile, but in the long run it could be introduce a trend for the computerization of the entire upward reporting procedure.

6.0 REFERENCES

1. Allen, P.E.T., and J.L. Masson: A Practical Method of Determining Availability and Accessibility of a Specific Forest Type Using Maps and Overlays, Forest Department of Burma, Food and Agriculture Organization of the U.N., Rangoon, September, 1982.

2. Bass, Jeff: Translating the SAS language into BASIC, Byte, Vol. 9, No. 1, January, 1984.

3. Bond, George : A Database Catalog, Byte, Vol. 9, No. 11, October 1984.

4. Brooks, Frederick P. : The Mythical Man-month, Addison Wesley Publishing Company,1975.

5. Calkins, Hugh W. : A Pragmatic Approach to Geographic Information System Design, Proceedings of a U.S./Australia Workshop held under the joint sponsorship of the U.S. National Science Foundation and the Australian Department of Science and Technology, International Geographical Union, Honolulu, Hawaii, 1982.

6. Carpenter, James , Dennis Deloria, David Morganstein: Statistical Software for Microcomputers, Byte, Vol. 9, No. 4,1984.

7. Case, Robert P. : Microcomputers in the Field, Byte Vol. 9, No. 6, June 1984.

8. Chamberlain, D.D. : Relational Data-Base Management Systems, ACM Computing Surveys, Vol. 8, No. 1, 1976.

9. Covvey, H. Dominic and Neil H. McAlister: Computer Choices, Microbooks, Addison Wesley Publishing Company, 1982.

10. Daniel, Terry C., William B. White, and Don O. Hunter: Integrated Pest Impact Assessment System: An Interface to a Forest Management Information System. United States Department of Agriculture, Forest Service, Forest Pest Management, Methods Application Group, Fort Collins Colorado, Report 84-2, 1983.

11. Date, C.J. : An Introduction to Database Systems, Vol I and Vol II, Addison Wesley Publishing Company, 1983.

12. Fry, J.P. and E.H. Sibley: Evolution of Data-Base Management Systems, ACM Computing Surveys, Vol. 8, No. 1, 1976.

13. Garrett, Lawrence, D., James J. Rogers, and Joseph M.Prosser:

Multiresource Inventory and Analysis Systems: Gaining Maximum Benefit for Land Managers, In-place Resource Inventories: Principles and Practices, Proceedings of a National Workshop, University of Maine, Orono, Maine, 1981.

14. Kay, Allan: Computer Software, Scientific American, September 1984.

15. Martin, James : Computer Data-Base Organization, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 07632, 1977.

16. Metzger, Philip W. : Managing a Programming Project, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1973.

17. Moser, J.W. : Illustration of a Generalized Data Base Management System For Maintaining Compartment Records. In Resource Data Management Symposium, Purdue University, Department of Forestry and Natural Resources, and Cooperative Extension Service, 1976.

18. Murphy, Dennis L. : Database Management: a Forest Inventory Application, United States Department of Agriculture, Forest Service, North Central Experiment Station, General Technical Report NC-67,1981.

19. Naisbitt John : Megatrends, Warner Books Inc., New York, New York, 1984.

20. Sibley, E.H. : The Development of Data-base Technology, ACM Computing Surveys, Vol. 8, No. 1, 1976.

21. Singh, K.D., and J.P. Lanly : Towards a Global Computerized Forest Resources Information System, FAO, undated paper.

22. Teichholz Eric, and Peggy Kilburn : Low cost CADD at Work, Datamation Vol. 29, No 1, January, 1983.

23. Wilson, Rodney , Hans T. Schreuder, and Brian M.Kent: A Description of Some Survey Sampling Computer Packages, In In-Place Resource Inventories: Principles and Practices, Proceedings of a National Workshop, August 9-14, University of Maine, Orono, Maine.

24. Space, J.C. : General Purpose Computer Programs to Process Forest Inventory Data in the United States. IUFRO National Forest Inventory Symposium, Institutul de Cercetari si Amenajari Silvice. Bucharest, Rumania.

7.0 COMPUTER JOURNALS

The following is a list of recommended computer journals. The stars following the journal name indicate the level of recommendation.

1. A+ (Apple), P.O. Box 2964, Boulder, Colorado 80321.

2. BYTE *** (all microcomputers), P.O. Box 59, Martinsville, New Jersey, 08836.

3. COMPUTER GRAPHICS WORLD *** (graphics and image processing), P.O. Box 122, Tulsa, Oklahoma 74101.

4. COMPUTERS AND ELECTRONICS (all microcomputers), P.O. Box 13877, Philadelphia, Pennsylvania 19101.

5. CREATIVE COMPUTING (all microcomputers), P.O. Box 5214, Boulder, Colorado 80321

6. DATAMATION*** (mini- and mainframe computers), P.O. Box 1043, Barrington, Illinois 60010-9978.

7. INFOWORLD** (all microcomputers), 375 Cochituate Road, Framingham, Massachusetts 01701.

8. MICRO (all microcomputers), P.O. Box 6502, Chelmsford, Massachusetts 01824.

9. MICROSYSTEMS*** (all microcomputers), P.O. Box 2937, Boulder, Colorado 80322.

10. NIBBLE (all microcomputers), P.O. Box 325, Lincoln, Massachusetts 01773.

11. PC MAGAZINE *** (IBM PC and PC compatibles), P.O. Box 2443, Boulder, Colorado 80321.

12. PC WORLD *** (IBM PC and PC compatibles)

13. POPULAR COMPUTING (all microcomputers), P.O. Box 307, Martinsville, New Jersey 08836.

14. PC TECH JOURNAL (IBM PC and PC compatibles), P.O. Box 2996, Boulder Colorado 80321.

15. SYSTEMS AND SOFTWARE*** (all computers), P.O. Box 1411, Riverton,

New Jersey 08077.

16. 80 MICRO (all microcomputers), P.O. Box 981, Farmingdale, New York 11737.

APPENDIX A

RASTER SYSTEM USE EXAMPLE

In this appendix an attempt is made to apply the functions of an existing raster based system (MAPS of the MOSS system) to a problem in Burma, reported by P.E.T. Allen and J.L. Masson (1982). This problem was previously solved by manual overlaying of maps. A study such as this could be used as part of a feasibility study, and could be of tremendous help in gaining an appreciation for the functionality of a system under consideration. It is best of course, to actually try the real system, but if this is not possible, one might carefully study the manual, and try to visualize how the commands may be used to solve the problem. The latter course of action was taken for this example.

A.1 SYSTEM FUNCTIONS

Reclassification Functions - The purpose of reclassification functions is to reassign or reclassify values of an existing map. As a result the boundaries inherently represented by the different gridcell values will change.

RENUMBER. This function assigns new values to existing values according to user specified reassignments. As a result, the boundaries that are implied by the cell values will change. Cells which are not reassigned keep their original value.

EXTRACT. Same as RENUMBER, but cells which are not reassigned become background values.

AGGREGATE. Takes multiple layers, and combines them into one layer.

MERGE. Combines two or more adjacent gridlayers into one gridlayer.

CATEGORIZE. Create a new discrete cell map by counting occurrences of each cell value in a continuous map.

CUT. Cut out a portion of a map by specifying top and bottom row, and left and right column.

FUNCTION. Reassign cell values according to a mathematical function such as the logarithm or the sine.

ISOLATE. Create multiple binary maps from single input maps.

SLICE. Divides a range of values into intervals, and assigns the interval number to the value in the interval.

Overlay Functions - One difference between reclassification and overlay functions is the number of layers. Reclassification functions operate on one input

layer, overlay functions operate on more than one input layer. The input layers are 'overlaid', and a new grid layer is created of which the cell values are a function of the corresponding cell values of the input maps. Depending on the specified operation, the function may be a logical, arithmetical or statistical operation, or be a combination thereof. The overlay functions always operate on a cell by cell basis: for each cell the input values are transformed to one output value, usually with a simple arithmetic operation such as addition, or a logical operation such as 'or'. This basic operation is repeated for all cells in the map.

ADD. Adds two or more maps.

AVERAGE. Averages two or more maps.

BOOLEAN. Performs Boolean operations on binary maps. Boolean functions can be AND, OR, XOR (exclusive or) and NOT.

COVER. Covers one map with a second map. A portion of the second map may have zero values, where the first map will show through.

CROSS. Logical combinations of categories of input map values can be assigned an output value. For instance 1-50 on map 1, and 30-80 on map 2, is assigned the value 1, 50-100 on map 1, and 0-30 on map 2, are assigned a value of 2, while the remainder is set to 0.

DIVIDE. Divide one map by another map, and the result by a third map, and so on.

MATH. Performs arithmetic and algebraic functions on a number of maps.

MAXIMIZE. Assigns the largest value of the input maps to the output maps.

MINIMIZE. Assigns the smallest values of the input maps to the output map.

MULTIPLY. Multiplies one map with another, and the result with a third map, and so on.

SUBTRACT. Subtracts two or more maps.

Note that these functions are by no means mutually exclusive: there are many functionality overlaps. For instance, the MULTIPLY, DIVIDE, SUBTRACT and ADD functions can also be performed by MATH. The reason for this overlap, and a possible different arrangement is discussed in the section on the user interface in chapter 2 of this report.

Distance Functions - Distances are used in this class of functions. For instance, a buffer zone around a linear feature may be created, in which all cells that are within a certain distance from that feature are 'turned on'. Distance is a rather flexible concept: it does not always mean straight line distance. It may be the cost along a path, and the path may not be straight.

This type of function usually operates within a single map, but at each step multiple cells are involved in the calculation of a single output cell.

ZONE. This function designates a group of cells origin cells, and then generates a buffer zone around these cells. The user can specify the origin cells and the zone width by which the zone is 'grown' around the origin cells. The zone can be divided into rings. Each cell in the zone is assigned a value equal to the ring number plus one. The first ring is closest to the origin cells, which are assigned a value of one.

PROXIMITY. A set of origin cells is selected on the first map by specifying a value range, and a set of target cells is similarly selected on the second map. If the target cells lie within a certain distance of the origin cells, than the target cells are copied to the output map.

Neighbourhood functions - This is the last class of analysis functions. The cells in the neighbourhood of a cells position are used to compute the value for that cell in the output map. Typically, a roving 'window' is passed over the map, and the center cell of the window receives the value computed on the cells in the window. Maps derived from digital terrain models are frequently made with this technique. Another type of map derived from neighbourhood operations is the diversity map, in which the output map is an index of the variability of the input map.

ASPECT. The direction of the surface slope is computed from a window containing elevations.

SLOPE. The surface slope for each cell is computed from a window containing elevations.

SCAN. A new value for the center cell of a window is computed according to a user selected criterion. User options are: average, total, maximum, minimum, median, least frequently occurring, number of different values (diversity), etc.

VISTA. The user selects a viewpoint, and the function returns a map indicating what portions of the map are visible and invisible from this viewpoint. The function operates on a map with elevations, and is concerned with the entire map, rather than with a number of small local windows.

A.2 BAMBOO EXAMPLE

To give an example of the potential use of a raster based system for solving a real problem that may occur in a developing country we will attempt to apply some of the functions of the previous section to a problem of determining the availability and accessibility of bamboo in the southern area of the Arakan Forest Division by using maps and transparent overlays. We will attempt to see how the same problem might have been approached if a raster based GAS had been available.

The project was requested to complete a feasibility study for the proposed

establishment of a pulp and paper mill in the Kyeitali area of the Arakan Forest Division using bamboo (Melocanna bambusoides) as the raw material.

To assess the supply situation for such a mill, it was important to prepare a final map showing the major areas of bamboo with superimposed delineations indicating the three major utilisation-accessibility classes: easy, reasonable to difficult, very difficult. This map could then be superimposed on a map of reporting units with known bamboo volumes by reporting unit, to obtain a final tabulation of utilisible volume by reporting unit.

First we will attempt to summarize the approach used by Allen and Masson, taking some poetic license to restructure the sequence of events for the sake of clarity, and then we will speculate how the final maps might have been built on a digital system. Note that we are not attempting to promote the digital approach; the practicality of each method is entirely dependent on the circumstances under which the analysis must be carried out. We will however give a brief synopsis of the advantages and disadvantages of each method at the end of the example.

The following sources of map information were available for the project:

1. Definition of the area of interest

2. Topographical maps, last updated in the 1940's with the aid of aerial photography

3. Aerial photographs; the most recent coverage at a scale of 1:25,000

4. Stereograms of different landforms and forest types

5. One Landsat scene

6. Inventory data of a recent inventory with plots laid out at 6 km intervals

Non-map information that was available:

1. Management plans

2. Precipitation figures and climatic data

For convenience we will divide the subsequent processing of these information sources into five stages, each with a number of output maps. To avoid confusion we would like to point out that a basemap contains several themes, while we will use the term map for a single theme map or overlay.

In the first stage of the project, the following basic maps were constructed:

1. A revised basemap. This map shows the main drainage pattern, roads and tracks, railways and areas of habitation. It was derived from the

topographical maps, mainly by simplifying the drainage pattern.

2. A detailed forest type map derived by photointerpretation using the available stereograms

3. A map showing the plot locations and their forest types

4. A contour map

5. A map of reporting units

In the second stage the detailed forest type map was simplified and combined with the inventory plot map. Four major vegetation types remained: bamboo, mixed deciduous, bamboo and mixed deciduous, and mixed deciduous and bamboo. A choice between the latter two classes was based on the predominance of the two major types. Bamboo was only reported when occurring with a groundcover of more than 60%.

Also in the second stage, the slope class map was processed into an 'internal accessibility map'. Internal accessibility was defined as the degree of ease by which material felled in the forest area can be transported to the nearest landing for transhipment by road or water to its final destination. Degree of slope was considered the major factor for this kind of accessibility, and a slope class map was constructed using a manual method, with the following slope classes: < 8%, 8-12%, 12-20%, > 20%. Products resulting from the first stage were therefore:

1. A modified forest type map

2. An internal accessibility map

The objective of the third stage was to prepare a relative accessibility map. Relative accessibility was defined as the accessibility for larger areas, specifically as pertaining to accessibility from the forest to the main landing or mill site.

At first it was thought that the products from first and second stage would suffice for the management officer to make a relative accessibility map. However, the modified basemap apparently still contained too much detail, so that a special map was prepared with only potentially floatable rivers and chaungs. At this point Allen and Masson also realized that additional information was required, namely a major and minor watershed divides overlay, and information on geology and climatology. Additional second stage map products were therefore:

1. A rivers and chaungs map

2. A major and minor divides map

The geology and climatology information was mainly used to establish that road construction would be difficult, so that extraction would have to take place by rafting,

and to establish optimum time periods for this activity.

With this information a relative accessibility classification map was made as the third stage of the project. Sixteen classes were recognized as combinations of the four slope classes and four accessibility classes, namely: accessible, accessible by water, difficult of access, and economically inaccessible. For instance, A2 would have the following meaning: $< 8\%$ slope, accessible by water, likewise C3 would mean 12-20% slope, difficult of access.

The exact procedure for arriving at the relative accessibility classification is not presented in Allen and Masson (1982), and we will therefore make some assumptions in presenting the digital case.

In the fourth stage the relative accessibility map was combined with the modified forest type map of the second stage, to produce the final map. This map was simplified to contain four classes namely: non-bamboo, bamboo easy, bamboo reasonable to difficult, bamboo very difficult.

In the fifth and final stage the final map was overlaid on the map with the reporting units to obtain the areas of each of the classes by reporting unit. Knowing the bamboo volume per ha for each unit, the volume by final class could then be reported by unit.

Using hindsight, perhaps a better sequence of steps can be defined, but to be realistic we will try to use the same stages and steps of the manual approach.

At the beginning of the digital project we assume that the following digital maps are available, these correspond to the maps produced at the end of stage one of the manual project:

1. A map with roads, tracks and railroads (discrete map)

2. A forest type map (discrete map)

3. A plot location map (discrete map)

4. An elevation map (continuous map)

5. A drainage map (discrete map)

Unlike the manual project, all these maps will be used as single theme maps, and for this reason roads and tracks have been separated from drainage. All maps are digitized and rasterized. The elevation map is a special case. Contours are digitized, and the contours are converted to a continuous elevation map with special software not available in the MAPS system. The roads, tracks and railroad map and the drainage map show these linear features, and for the drainages special care has been taken to code major and minor streams with different cell values. The plot map show the plots as single cells with values corresponding to the forest types of the forest type

map.

The first step of the second stage is to modify the forest type map by combining it with the inventory map. The objective here is to reconcile the two maps, when classifications are different between photointerpretation and inventory. Unfortunately, it appears that this cannot be accomplished with the functions described for the MAPS system. In an other system known to the author it is possible to look at the combinations of attributes that occur between maps, and these combinations can then be reasssigned on the original maps. With the current system one must go back to the rasterization in the beginning and manually reassign type values.

The second step of the second stage is to generate a slope class map and turn the slope class map into an internal accessibility map. With the available elevation map this is accomplished with the SLOPE command. This produces a continuous map with percent slope for each cell. To reclassify this map into a discrete relative accessibility map with the four slope classes the RENUMBER command can be used with the following command phrase:

```
RENUMBER slopemap FOR internalaccessmap
ASSIGNING 1 TO 0 THROUGH 7
ASSIGNING 2 TO 8 THROUGH 11
ASSIGNING 3 TO 12 THROUGH 20
ASSIGNING 4 TO 20 THROUGH 500
```

In the third stage, to make a relative accessibility map, Allen and Masson realized that the drainage net was to detailed and that a map with major and minor divides was needed.

With a digital drainage map, with codes say of 1 for major rivers and chaungs and 2 for minor streams, this would simply we a matter of reclassification as follows:

```
EXTRACT drainagemap FOR majordrainagemap
ASSIGNING 1 TO 1
```

With the EXTRACT functions, unassigned cells become background, and so the minor streams disappear.

Instead of making a map of major and minor divides, for the digital analysis it is more convenient to make a map of the watersheds relevant to the bamboo mill. We will therefore assume that such a map is constructed, digitized and rasterized and coded with 1 for relevant watersheds and 0 for the remainder of the area.

A map with 16 different accessibility classes was created by Allen and Masson, based on the four slope classes, and four accessibility classes, namely: accessible, accessible by water, difficult of access, and economically inaccessible. As they do not explain how these classifications were interpreted, we will give them the following meaning: accessible - within one kilometer of road, railroad or track; accessible by

water - within one kilometer of a major river or chaung ; difficult of access - removed more than one kilometer from any road, track, railroad, river or chaung; economically inaccessible - in the wrong watershed.

This interpretation suggests the use of the ZONE function to generate 1 km zones around roads, railroad, and tracks with the map containing these features using the command phrase:

ZONE roadsmap INTO 1 TO 1000 FOR roadzones

Similarly this is done for the major rivers and chaungs with the phrase:

ZONE majordrainagemap INTO 1 TO 1000 for riverzones

Zones in both maps are assigned a value of 2, background cells will have a value of zero.

Now, to create a map with the first three accessibility classes, we can use the CROSS function as follows:

CROSS roadzones WITH riverzones
ASSIGNING 2 TO 0 THROUGH 2 AND 2
ASSIGNING 1 TO 2 AND 0 THROUGH 2
ASSIGNING 3 TO 0 AND 0
FOR temporarymap

This makes a combined map of road and river zones, with road access overlaid on water access, where the zones overlap (this is why the statement ..ASSIGNING 1..comes after ..ASSIGNING 2..). Thus road access will have priority over water access.

A second CROSS of the temporary map with the watershed map will produce a map with all four accessibility classes:

CROSS temporarymap WITH watershedmap
ASSIGNING 1 TO 1 AND 1
ASSIGNING 2 TO 2 AND 1
ASSIGNING 3 TO 3 AND 1
ASSIGNING 4 TO 1 THROUGH 3 AND 0
FOR fourclassmap

The effect of this cross is simply to change all zone values outside the relevant watersheds to class 4 (economically inaccessible),

To obtain a map with the full 16 classes, one must do yet another CROSS as follows:

```
CROSS fourclassmap WITH internalaccessmap
ASSIGNING 1 TO 1 AND 1
ASSIGNING 2 TO 2 AND 1
ASSIGNING 3 TO 3 AND 1
ASSIGNING 4 TO 4 AND 1
ASSIGNING 5 TO 1 AND 2
etc....
FOR relativeaccessmap
```

In the fourth stage the 'final map' is created by combining the relative accessibility map with the forest type map. With the digital method, this is again a use of the CROSS function with a simultaneous reassignment of the forest types into bamboo and non-bamboo, and a simplification of relative access into three classes, namely easy, reasonable to difficult, difficult. For instance, class 2 of the relative accessibility map means accessible by water on slopes less than 8%,and so this can be reclassified as class 1: easy. Taking classes 1,2,5 and 6 as easy (access by road or water and slopes less than 12%), the first part of the CROSS phrase for the bamboo-easy would read (assuming that the first two types in the forest type map are bamboo classes):

```
CROSS finalmap WITH foresttypemap
ASSIGNING 1 TO 1 THROUGH 2 AND 1 THROUGH 2
ASSIGNING 1 TO 5 THROUGH 6 AND 1 THROUGH 2
...etc.
FOR extractionmap
```

In the fifth and last stage, the extraction is overlaid on the reporting unit map to obtain areas of the final extraction classification by reporting unit. Again, this exercises the CROSS function, assigning a class to each reporting unit - extraction type combination. Areas of each combination can then be measured with the AREA function (one of the MAPS display functions, and not earlier discussed). Area figures can then be multiplied with bamboo volumes for each reporting unit to find the volume figures for each extraction class by reporting unit.

APPENDIX B

SYSTEM DESCRIPTIONS

B.1 INTRODUCTION TO SYSTEM DESCRIPTIONS

Name, vendor, vendors address and telephone number are provided for each system. This will be followed by a brief description of the hardware.

An entry level system is considered where appropriate. Such a system is adequate to get started, with a capability to solve realistic problems but otherwise on the lower end of the range with respect to capacity, number of peripherals computing speed, etc.

A price category will also be stated. The reader should be aware that price quotations have been obtained through telephone conversations with the vendor, and that actual written quotes may be substantially different. The listed proces should be considered as 'ballpark' figures.

If known, installations in developing countries will be mentioned.

B.2 SYSTEM DESCRIPTIONS

B.2.1 System 2000

- *System name*: System 2000

- *Vendor*: INTEL Commercial Systems Division

- *Address*: P.O. Box 9968, 1275 Research Boulevard, Austin, Texas 78766, U.S.A.

- *Telephone*: (512) 258-5171

- *Telex*: none

- *Price Category*: $60,000 (one time license purchase)

- *Hardware Configuration*: IBM System/370 class, Sperry Univac 1100 Series, CDC 6000/Cyber series.

- *System Description*: A logically structured hierarchical database management system. The system is driven from a host language. The hierarchical structure consists of a set of indices that supply the data to data links and logical views of the data. Tabular rows and columns describe the structure; a row is a record, and a named column is a component. There can be maximally 1000 components and up to 16.7 million records. There is a

DDL called DEFINE, and an interactive data definition facility called CREATE. There are two DML's: QUEUE and ACCESS. The first is suitable for interactive applications, while the second is oriented towards batch jobs. There is an integrated Data Dictionary that can drive THE DEFINE DDL. Other DBMS facilities: accounting, rollback/recovery, and security through password controls, even at the data item level. There are a variety of special features such as an concurrent processing capability, a query language simulating relational capabilities, as well as report writers and graphics capabilities.

- *Installations in developing countries*: the system has a worldwide distribution.

B.2.2 SEED

- *System name*: SEED
- *Vendor*: Seed Software
- *Address*: Suite 217, 2300 Walnut Street, Philadelphia, Pa 19103, U.S.A.
- *Telephone*: (215) 568-2424
- *Telex*:
- *Price Category*: $20,000-40,000 (one time license fee, depending on the hardware).
- *Hardware Configuration*: IBM system/370,3000,4300,CDC 6000 and Cyber, DEC DEC system 10/-20, VAX, PDP, Hewlett Packard HP 3000, Perkin-Elmer 3200, SEL 3227/77, Prime 50 series, Modcomp.
- *System Description*: A network system with some hierarchical features. Data items (attributes) are grouped into records, which are grouped into sets. Sets have an owner record and a chain of linked members. Each record can participated as an owner or member in up to 35 hierarchies. One-to-one, one-to-many, and many-to-many relationships can be defined. It has a DDL for both the schema and the subschema. The DDL is used to establish the network of pointers that interrelate stored data, and makes the relationships known to the user. The DML provides a mixture of batch and interactive capabilities. Host languages can be FORTRAN or COBOL. Statistical reports can be produced with DBSTAT. The systems has security through passwords down to the data item level, as well as other integrity features such as journaling and transaction roll back. Other special features are: GARDEN, HARVEST, BLOOM, and SPROUT. GARDEN provides an interactive environment with Help facilities, HARVEST is an interactive query language (DML), BLOOM is a report writer, and SPROUT is a transaction handler.

- *Installations in developing countries*: international distribution.

B.2.3 RIM

- *System name*: RIM

- *Vendor*: Boeing Computer Services Company, Software and Education Products Group.

- *Address*: P.O. Box 24346, MS 7A-20, Seattle, Washington 98124

- *Telephone*: (206) 763-5185

- *Telex*:

- *Price Category*: $14,000

- *Hardware Configuration*: CDC Cyber, IBM (VM/CMS), DEC VAX (VMS), CRAY (COS). Under consideration are: CDC Cyber (NOS/BE), IBM (MVS/TSO), Sperry UNIVAC (Exec 8), Prime (Primos), DEC VAX (Unix), Data General MV-8000

- *System Description*: A relational system that is not strictly a DBMS because it does not allow for multiple users, but it has all the other required features. The database is organized into two- dimensional tables of rows and columns. It provides facilities to load, combine, sort and retrieve data. It support relations with an unlimited number of rows (within disk capacity), an unlimited number of attributes total, but a limited number of attributes (columns) per relation. Data types supported are: text, real, integer, double precision, maxtrix, vector, date, currency, and time. Relational operators supported are: project - extracts and sorts attributes from selected rows to create a new relation - ; intersect - matches values of common attributes between two relations to create a new relation - ;join - satisfies comparison criteria between two specified attributes of two relations to create a new relation - ; subtract - forms a new relation where common attributes do not match between two relations - ; union - combines rows in two relations having matching common attributes, then adds rows without matching attribute values and flags missing values - ; select - retrieves rows in one relation where boolean combinations of attributes are true -. Attributes can be computed from other attributes. Data constraint rules for data loading can be defined. There is full password protection at the database and relation levels. Attributes maybe 'keyed' for fast indexed access. There is an interactive as well as a FORTRAN programming interface. A report writer can be purchased separately. A graph plotting capability is included.

- *Installations in developing countries*:

B.2.4 DBASE II

- *System name*: dBASE II
- *Vendor*: Ashton-Tate
- *Address*: 9929 West Jefferson Boulevard, Culver City, Ca 90230, U.S.A.
- *Telephone*: (213) 204-5570
- *Telex*:
- *Price Category*: $500
- *Hardware Configuration*: 8080-,8085, or Z80 based microprocessor systems with at least 48K bytes of memory, one or more floppy disk drives, and a cursor addressable character CRT.
- *System Description*: A relational-like data management system, in which a database is a similar to a relation (two-way table) in a real relational DBMS. At most two databases can be accessed simultaneously, but there are no relation operators such a join or intersect. Data in a database can be selectively retrieved using arithmetical and logical operators. The system uses English like commands for interactive access and has its own programming language with IF..THEN..ELSE and DO..WHILE and other looping and branching constructs. It supports up to 65535 records per database file, up to 1000 characters per record, and 254 characters per attribute field. There are commands to edit data (edit, replace, change, delete, recall, pack), to add data (append, create, insert), and to display data (display, list, report, read, sum, total), as well as file manipulation commands and other miscellaneous commands.
- *Installations in developing countries*: International Distribution, and more than 10,000 installations.

B.2.5 MicroRIM

- *System name*: Micro-RIM
- *Vendor*: MICRORIM Inc.
- *Address*: 1750 112th N.E. Suite A200 P.O.Box 585, Bellevue, Washington 98009
- *Telephone*: (206) 453-6017
- *Telex*:

- *Price Category*: $1400

- *Hardware Configuration*: Z80,8080,8085, Apple II or III with Z80 card, IBM Personal Computer, Victor 9000, Burroughs B20, and Convergent Tech Computer. Operating systems: MS-DOS, CTOS, CP/M. Z80,8080,8085 require 65K-byte memory, other systems 256 K-bytes of memory. Sufficient disk storage is required.

- *System Description*: Microcomputer version of RIM (described earlier). Maximum of 20 relations, number of rows limited by disk capacity. Can have up to 400 attributes over all relations, maximum rowsize 1274 bytes. Data types supported are: dates, currency, 9 digit integers, real values with 7 digit accuracy, scientific or decimal notation, text, and time. Has most of the minicomputer RIM functions, as well as others not available with the larger RIM. Has customized form-creation report writer.

- *Installations in developing countries*:

B.2.6 Knowledgeman

- *System name*: Knowledgeman

- *Vendor*: Micro Data Base Systems Inc.

- *Address*: POB 248, Lafayette Indiana 47902, USA.

- *Telephone*: (317)463-2581

- *Telex*:

- *Price Category*: $500. Options: graphics ($225), screen print ($100), mouse support($100), run time support ($100).

- *Hardware Configuration*: Hardware with CP/M-86, PC-DOS or MS-DOS operating systems, 192 Kilobytes of RAM.

- *System Description*: Integrated spreadsheet, statistics, printed forms management, SQL-like inquiry, screen I/O management, structured programming language. Has graphics option. Allows 65535 maximum characters per field, 255 maximum fields per record,and an unlimited number of open files.

- *Installations in developing countries*:

B.2.7 Minitab

- *System name*: Minitab

Vendor: Minitab Inc. 215 Pond Laboratory

Address: University Park, Pennsylvania 16801, U.S.A.

Telephone: (814) 865-1595

Telex: 84-2510

Price Category: $1000-$1800 yearly lease.

Hardware Configuration: CDC/NOS, DG/AOS/VS, DEC PDP-11, VAX/VMS/RT-11, Prime /Primos, and other computers.

System Description: Written in FORTRAN IV, FORTRAN 77 source available. Originally developed for an introductory pre-calculus statistics course given at Pennsylvania State University. Provides excellent introduction to the use of others packages such as SPSS, BMD,SAS. Functional capabilities fall in the following categories: entering data, outputting data, editing and manipulating data, arithmetic on data, column and row operations, plots and histograms, basic statistics (t tests, correlation), regression, analysis of variance (one-way, two-way), nonparametric statistics, tables, time series, exploratory data analysis. The systems features transformations and coding as well as automatic handling of missing values.

- *Installations in developing countries*: More than 1000 systems installed.

B.2.8 SPSS

System name: SPSS X

Vendor: SPSS Inc., Marketing Department.

Address: Suite 3300, 444 North Michigan Avenue, Chicago, Illinois, 60611, U.S.A.

Telephone: (312) 329-2400

Telex:

Price Category: available by lease only: $8000 for first year, $5000 per year thereafter (international pricing). Maintenance, updates and consultation included.

Hardware Configuration: IBM/OS, DEC VAX, DG/AOS/VS, Honeywell/GCOS

System Description: Written in FORTRAN. Offers advanced file and data management capabilities, expanded data displays, and the capability to accept input in any form. System is batch oriented. It has the following

capabilities: input from almost any type of data file; file management, including sorting, splitting, and aggregating files, match-merging multiple files, and saving fully defined system files; data management, including sampling, selecting, and weighting cases, recoding variables, and creating new variables using numeric and string functions; tabulations and statistical analysis, report writing, and device independent graphics. Major categories of statistical analysis are: t-tests, ANOVA (one way and general linear models), loglinear analysis (including logit analysis), scattergrams, Pearson correlation, partial correlation, regression, discriminant analysis, factor analysis, nonparametric correlations, nonparametric tests, Box-Jenkins tests (time series analysis) reliability and survival models. SPSS X is a batch system. It can be executed interactively, but lacks the required error handling. SCSS is the conversational version of SPSS X.

- *Installations in developing countries*:

B.2.9 BMDP

- *Vendor*: BMDP Statistical Software Inc.

- *Address*: 1964 Westwood Blvd., Suite 202, Los Angeles, California 90025, U.S.A.

- *Telephone*: (213) 475-5700

- *Telex*: 499-2203 BMDP SOFTWARE

- *Price Category*: yearly lease for $2200-2900.

- *Hardware Configuration*: CDC Cyber, Honeywell, Univac 1100, Univac 70/90 PDP-11,Hp-3000,Riad 20.

  System Description: BMDP programs originated in the late fifties; the first manual was published in 1961. The software was first developed by the Department of Biomathematics of the School of Medicine of the University of California at Los Angeles. As such the package has more 'hard core' statistical functions than other systems. Major functional categories are: data description, frequency tables, regression analysis, analysis of variance, multivariate analysis, life table and survival analysis, cluster analysis, nonparametric analysis, analysis of variance, and covariance. Regression analysis covers both linear and derivative free non-linear analysis.

  Installation in developing countries:

B.2.10 SAS

- *System name*: SAS
- *Vendor*: SAS Institute Inc.
- *Address*: SAS Circle, Box 8000, Cary, North Carolina 27511, USA.
- *Telephone*: (919) 467-8000
- *Telex*: 802505
- *Price Category*: $6000 annual fee for non-tranferable, non-exclusive license for SAS, and SAS/GRAPH. Maintenance, updates, and consultation included.
- *Hardware Configuration*: Available for IBM, DEC VAX, DEC-20, DG AOS/VS, PRIME.
- *System Description*: SAS has the following utility capabilities: data input, data transformations (character and numeric), merging, matching, and sorting of files, matrix manipulations (extensive package, but not available under VAX VMS), sample amalgamation, agregation. Statistical capabilities include the following categories: descriptive statistics, regression, analysis of variance, categorical data analysis, multivariate analysis, principal components analysis, discriminant analysis, clustering, scoring. Other SAS packages (not yet available on DEC VAX) are SAS/ETS for modeling and simulation, SAS/FSP for full-screen data entry and edit, SAS/OR for operations research, SAS/IMS for interface with IBM IMS DBMS.
- *Installation in developing countries*:

B.2.11 ABSTAT

- *Vendor*: Anderson-Bell
- *Address*: P.O. Box I91 Canon City, Colorado 81212, U.S.A.
- *Telephone*: (303) 275-1661
- *Telex*:
- *Price Category*: $400
- *Hardware Configuration*: hardware with any of the following operating systems: CP/M-80, CP/M-86, PC-DOS, 56 or 128 Kilobytes of RAM required, as well as two disk drives.
- *System Description*: Easy to use general statistics package with good

error handling. Has good data management and data processing capabilities. Statistics of the type found in elementary textbooks. ANOVA allows only one observation per cell. A new version will be available in early 1984. The only micro system with an interface to a database package: dBase II.

- *Installation in developing countries*:

B.2.12 Dipix

- *System name*: ARIES II
- *Vendor*: Dipix Systems Limited
- *Address*: 1785 Woodward Drive, Ottawa, Ontario, Canada K2C 0P9
- *Telephone*: (613) 224-5175
- *Telex*: 0533946
- *Price Category*: $100,000 (export)
- *Hardware Configuration*: Dynamic Image Display Subsystem (Dipix hardware), 2 Megabit Image Memory, LSI 11/23 processor, Ampex Winchester disk, operator console, tablet, colour TV monitor.
- *Expandibility*: The system can grow modularly. Larger processors: PDP, VAX. Entry level system can be used as workstation. Image memory expandable to 16 Megabit. Also available : 330 Megabyte disk drives. Possible add ons: floppy disk, magnetic tape, image scanner, film recorder, colour plotter, and map digitizer.
- *System Description*: Mostly IAS system. Some GAS capabilities, through functions for logical map combining (and, exclusive or, etc.) as well subtraction and ratio's of images. Vendor claims that enough 'hooks' are available to interface with other GIS software. The system does not have a vector to raster conversion function, but work is underway to create an Intergraph interface for the Intergraph exchange format (SIF).
- *Installation in developing countries*: China (5), Indonesia (3),Thailand (2), Peru (1).

B.2.13 ESRI GRID

- System name: GRID, GRID/TOPO
- *Vendor*: Environmental Research Systems Institute (ESRI)

- *Address*: 380 New York Street, Redlands California 92373 USA

- *Telephone*: (714) 793-2853

- *Telex*: TWX 910 332 1317

- *Price Category*: $54,900 (export, travel and per diem not included)

- *Hardware Configuration*: This is a software system written in FORTRAN IV, and requiring less than 256 Kilobytes. It is therefore transportable to most mini and larger computers. The system is most prominently used on Prime and Vax equipment.

- *System Description*: This is ESRI's raster based system. It is a typical GAS system: GRID deals with deals with thematic data, and GRID/ TOPO handles continuous terrain data. Two major programs of interest are the grid-modelling program and a corridor analysis program. The first allows the combination of maps with boolean logic and arithmetic and the second allows for optimal routing through a raster with cost values. Other functions compute minimum distances, weighted and unweighted numbers of occurrences, least radii, buffer zones, etc. Among the topographic functions are slope and aspect calculations. The user interface for the grid-modelling program is of the language type, with a rather rigid syntax.

- *Installation in Developing Countires*: Venezuela (2)

B.2.14 ERDAS

- *System name*: ERDAS 2300,2400

- *Vendor*: Earth Resources Data Analysis Systems

- *Address*: 99 McMillan Street NW, Atlanta Georgia 30318 USA

- *Telephone*: (404) 872-7327

- *Price Category*: $96,000 + 12-30% for export, depending on country

- *Hardware Configuration*: PDP 11/23+, 2 10 Megabyte hard disks, 9 track tape drive, image processor, RGB (Red,green, blue) monitor, joystick, DEC VT100 terminal, graphics matrix printer, cabinetry.

- *Expandibility*: ERDAS 2400 has PDP 11/24 CPU for a total system cost of approximately $120,000.

- *System Description*. This system is a IAS/GAS system. It comes with a full Landsat processing capability and a fully integrated set of GIS type

functions. A polygon based digitizing system is included. Typical GIS functions are: overlay and proximity analysis, pairwise combination analysis, overlay with weight factors. There is no specific attribute handling facility. The company also manufactures two micro based systems, however for support reasons it has decided not to export the ERDAS 100 system, and is currently undecided about the ERDAS 400. The latter two systems are built around Z80 microprocessors.

- *Installation in developing countries*: Panama (1). This is a U.S. Government system.

B.2.15 International Imaging Systems

- *System name*: Model 75 Image Processor

- *Vendor*: International Imaging Systems

- *Address*: 1500 Buckeye Drive, Milpitas, California 95035, USA

- *Telephone*: (408) 262-4444

- *Telex*: 172854 I2S MPTS

- *Price Category*: $100,000 (for export add 12%)

- *Hardware Configuration*: Model 75 image processor. This processor can be interfaced to other host computers such as DEC PDP-11, VAX-11, Data General Eclipse, Hewlett-Packard 1000 or 3000, or it can be configured with its own processor (LSI 11/23 with RSX 11-M, or MASSCOMP MC68000 with Unix; the listed price category is for the latter configuration). Four memory channel pairs are included for an entry level system (one pair consists of 512x512x8 bits). Other components: a color CRT, a man-machine interface consisting of a small keypad with a trackball and a footswitch, DEC VT100 terminal.

B.2.16 ESRI GRIDAPPLE

- *System name*: GRIDAPPLE

- *Vendor*: Environmental Research Systems Institute (ESRI)

- *Address*: 380 New York Street, Redlands California 92373 USA

- *Telephone*: (714) 793-2853

- *Telex*: TWX 910 332 1317

- *Price Category*: $2000 for software (independently purchased hardware

approximately $8000), turnkey system $12,000, for export add 10-20%.

- *Hardware Configuration*: Apple II, with two floppy disk drives, and a Corvus 5, 10 or 20 Megabyte hard disk. Epson MX-80 dot matrix printer, Summagraphics or Calcomp digitizer board, color monitor.

- *System Description*: this is a version of ESRI GRID, GRID/TOPO installed on an Apple computer. A limiting factor is the size of the Apple colour display (40 x 40, high resolution mode cannot be used because of a six colour limit), the map must therefore be inspected piecewise with a panning function.

B.2.17 SPEC-DAT

- *System name*: RIPS (Remote Information Processing System)

- *Vendor*: Spectral Data Corporation

- *Address*: 112 Parkway Drives, Hauppage, New York 11788, USA

- *Telephone*: (516) 543-4441

- *Telex*: TWX 510-226-1481

- *Price Category*: $18,000 (not considering export)

- *Hardware Configuration*: CPU with following components: S-100 bus, Z-80A CPU board, 64K RAM Board, 16K RAM board, Disk controller board, I/O board, Joystick I/O board, Video imaging system boards. Other components: joystick, terminal console, colour display, 2 floppy disk drives.

- *System Description*: This system was originally developed at the EROS Data Center, Sioux Falls, USA. The commercial version is now manufactured by Spectral Data Corp. The system is fully compatible with the U.S. Government furnished software, to be obtained for $100. Spectral Data also has its own software, including a parallelepiped classifier, that is a subset of the Government software. A variety of peripherals can be obtained with the RIPS system. Interfacing is facilitated through the use of an S-100 bus. Basic image data are stored on dual density 8' floppy disks, which can hold the equivalent of 20 7.5' United Stated Geological Survey Maps at Landsat MSS Resolution. The U.S. Government software is quite diverse: in addition to a set of image processing functions, a number of GIS type functions are also available.

B.2.18 Swedish Space Corporation

- *System name*: EBBA (Enkel BildBearbetnings Apparat)

- *Vendor*: Svenska Rymdaktiebolaget

- *Address*: Tritonvagen 27, 17514 Solna, Sweden

- *Telephone*: 08-98-0200

- *Telex*: 17128 Spaceco S

- *Telegram*: Spacecorp Stockholm

- *Price Category*: unknown to author

- *Hardware Configuration*: the system is based on a Metric85 computer, developed by Scandia Metric AB. It includes a display memory with three eight bit image planes and four binary graphics planes for a 256 x 256 display size. Results are displayed on a colour monitor. The Metric85 uses a Z80 CPU. Also included are a 'tangentbord' with 97 'tangents' as well as disks units, and 62-192 Kilobytes of dynamic RAM.

- *System Description*: This is an image processing system, but a number of logical operations can be used (AND, OR, NOT) for GAS applications.

B.2.19 ESRI ARC/INFO

- *System name*: ARC/INFO

- *Vendor*: Environmental Research Systems Institute (ESRI)

- *Address*: 380 New York Street, Redlands California 92373 USA

- *Telephone*: (714) 793-2853

- *Telex*: TWX 910 332 1317

- *Price Category*: $195,000 (for export add approximately 40%)

- *Hardware Configuration*: Prime 2250 'Rabbit' CPU, 158 Megabyte disk, tape drive, printer, plotter, graphics CRT, 3 alphanumeric CRT's, digitizing tablet. modem.

- *Expandibility*: Upward compatability with other Prime CPU's (450,550, 650,750,850,950). DEC VAX version of the system is also available.

- *System Description*: arc/node vector based system consisting of a spatial component (arc), and a commercially available relational file management system (INFO by Henco). ESRI has numerous installations, some applied towards the management of extremely large areas. This

system replaces ESRI's earlier vector based PIOS system. The system has an overlay (AND,OR) capability, as well as as buffer zone and updating ('paste up') capability. A true GIS system. Many advances have been made in the past year, including an orientation towards 'production style' outputs. A networking and a map library capability are on the drawing board. No installations are in place in developing countries, but there will be one in Australia in the next six months.

B.2.20 Autometric

- *System name*: MOSS

- *Vendor*: Autometric Inc.

- *Address*: 2629 Redwing Road, Ft Collins, Colorado 80526

- *Telephone*: (303) 226-3282

- *Telex*: none

- *Price Category*: $94,000 (within USA, export difference depends on country)

- *Hardware Configuration*: Data General S20 16 bit CPU, tape drive, 140 Megabyte Winchester disk, Altek digitizer, 4 alphanumeric terminals, Zeta 3653 SX or Calcomp 965 plotter.

- *System Description*: This system has been developed by the U.S. Fish and Wildlife Service, Ft Collins, Colorado, USA. It is widely used by agencies of the US Department of the Interior. The software is in the U.S. public domain, and is available free; the cost of the system is therefore entirely determined by the hardware. Autometric maintains the systems, mostly through U.S. government funded efforts, and installs systems at customer sites. MOSS is a vector based analysis system: it has no digitizing and editing capabilities. There are two companion digitizing systems: AMS (an Autometric system), and ADS (a system developed by the U.S. Bureau of Land Management). An enhanced cartographic output system (COS) has been developed by the U.S. Fish and Wildlife Service. They are also incorporating a raster capability into the system based on the Yale Map Analysis Package (MAPS). Sometimes the acronym MOSS refers to the conglomerate of packages: AMS/MOSS/MAPS/COS.

- *Installations in developing countries*: None, but a Vax version of the system is installed at the University of Graz, Austria. This version of the MOSS analysis system is based on the Data General code existing in the summer of 1983. The AMS digitizing part of this system is derived from a PDP code conversion.

B.2.21 Comarc Systems

- *System name*: COMPIS PLN, SIM, GDMS
- *Vendor*: Comarc Systems
- *Address*: 315 Bay Street, San Francisco, Ca 94133 USA.
- *Telephone*: (415) 467-1300
- *Telex*: TWX 9103727731
- *Price Category*: $145,000 (add 25% for export)
- *Hardware Configuration*: Data General 32 bit MV4000 CPU, 84 Megabyte Winchester disk, tape drive, 8 communication lines, system, console, floating point processor, 60 CPS printer, 8 pen plotter, 22x22 inch digitizer, Tektronix 4012. Any RS-232 graphic devise that supports Tektronix Plot 10 can be used. Supported plotters are ZETA 822 or 3653SX, or Calcomp 900 series.
- *Expandibility*: installations on DEC VAX and IBM equipment are also available. Upward compatability with larger Data General machines such as the MV8000.
- *System Description*: Software consists of a digitizing system (SIM), a geographic database management system (GDMS), and the polygon location and networking system (COMPIS PLN). The spatial data structures used are of the arc-node type. The system has specially integrated forest management software, not usually found in other systems (part of GDMS). The system was originally designed to provide in-house support for planning and environmental work. Quite a few systems are installed with forest products companies in the USA. System has capability for overlaying lines on polygons, and polygons on polygons (AND and OR). A report writer has been completed during the past year.
- *Installations in developing countries*: Indonesia 4, Malaysia 1.

B.2.22 Geobased Systems

- *System name*: GS-1000
- *Vendor*: Geobased Systems
- *Address*: 725 West Morgan Street, Raleigh, N.C. 27603
- *Telephone*: (919) 834-1313

- *Price Category*: $105,000

- *Hardware Configuration*: LSI 11/23 16 bit CPU, 20 Megabyte Winchester or removable platter disk, floppy disks, color graphics AED CRT, 36x48 digitizing table, VT100 alphanumeric terminal, LA120 printer, Houston instruments 36 inch plotter system.

- *Expandibility*: The vendor has also installed its system on larger DEC equipment such as the VAX-11 series.

- *System Description*: This is a GIS system with a strong digiting system. The spatial analysis packages are called STRINGS and SNIP. The system consists of several LSI 11/23 CPU's joined in a local network. The vendor prefers to customize the polygon overlay capability of SNIP. A tabular database system is also included.

- *Installations in developing countries*: Zaire 1, Saudi Arabia 1.

B.2.23 Forest Data Consultants

- *System name*: Landpak

- *Vendor*: Forest Data Consultants

- *Address*: 2855 Telegraph Ave Suite 203, Berkeley, California 94705, USA

- *Telephone*: (415) 549-2189

- *Price Category*: $150,000

- *Hardware Configuration*: Prime 2250 (RABBIT) CPU, Calcomp digitizing tablet, Zeta 3653SX 4 pen plotter, 2 alphanumeric terminals, Lear Siegler ADM with Retrographics board.

- *Expandibility*: Upward compatible with all Prime CPU's (450,550,650,750,850,950).

- *System Description*: A vector based GIS developed specifically for forestry applications. Has very flexible attribute handling and reliable overlay functions with all Boolean operators and combinations (AND, OR, NOT). Capable of operating on multiple layers in a single pass. User interface is through natural language query. Has buffer zone and terrain handling functions. Map blocks are rather small. Polygonal system, with unique arbitrary line digitizing method. The system also has a paste-up updating function. It has been designed for day-to-day forest management based on a 'transaction' concept.

B.2.24 Autometric

- *System name*: MOSS

- *Vendor*: Autometric Inc.

- *Address*: 2629 Redwing Road, Ft Collins, Colorado 80526

- *Telephone*: (303) 226-3282

- *Telex*: none

- *Price Category*: $40,000 (within USA, export difference depends on country)

- *Hardware Configuration*: Data General 20 microsystem, 1.5 Megabytes of main memory, 30 Megabyte disk, 15 Megabyte cartridge diskdrive for backups and file transfer, 24x36 digitizing table, small xy plotter, possibly an ink jet plotter (not yet decided), color monitor. The color monitor is an IBM personal computer with special graphics boards.

- *System Description*: This is the new MOSS micro system. The vendor recommends the use of such a system for developing countries, when the applications and purpose for a system are to be explored. When the situation becomes well defined, the system can be expanded to suit the purpose.

B.2.25 Comarc Systems

- *System name*: COMPIS II

- *Vendor*: Comarc Systems

- *Address*: 315 Bay Street, San Francisco, Ca 94133 USA.

- *Telephone*: (415) 467-1300

- *Telex*: TWX 9103727731

- *Price Category*: $35,000 (add 25% for export)

- *Hardware Configuration*: Data General 20 or 30 micro system, 84 Megabyte Winchester disk, tape drive, 8 communication lines, system, console, floating point processor (with DG 30), 60 CPS printer, 8 pen plotter, 22x22 inch digitizer, Tektronix 4012. Any RS-232 graphic device that supports Tektronix Plot 10 can be used. Supported plotters are ZETA 822 or 3653SX, or Calcomp 900 series.

- *System Description*: This is Comarc system on a Data General Micro.

B.2.26 Intergraph

- *System name*: Micro II
- *Vendor*: Intergraph Corporation
- *Address*: One Madison Industrial Park, Huntsville Alabama 35807
- *Telephone*: (205) 772-2000
- *Telex*: TWX 810-726-2180
- *Price Category*: $40,000-$60,000
- *Hardware Configuration*: Based on MicroVAX II.
- *Expandibility*: Will support up to four workstations.
- *System Description*: Has performance comparable to VAX 11/751 system. A standard deskside configuration includes 3 Megabytes of memory, floating point accelerator, two 100 Megabyte disks, 45 Megabyte cartridge tape, and a communications processor with Ethernet Interface. Not available until second quarter of 1985.
- *Installations in developing countries*:

B.2.27 Additional System:

- *System name*:
- *Vendor*:
- *Address*:
- *Telephone*:
- *Telex*:
- *Price Category*:
- *Hardware Configuration*:
- *Expandibility*:
- *Installations in developing countries*:
- *System Description*:

B.2.28 Additional System:

- *System name*:
- *Vendor*:
- *Address*:
- *Telephone*:
- *Telex*:
- *Price Category*:
- *Hardware Configuration*:
- *Expandibility*:
- *Installations in developing countries*:
- *System Description*:

B.2.29 Additional System:

- *System name*:
- *Vendor*:
- *Address*:
- *Telephone*:
- *Telex*:
- *Price Category*:
- *Hardware Configuration*:
- *Expandibility*:
- *Installations in developing countries*:
- *System Description*:

B.2.30 Additional System:

- *System name*:

- *Vendor*:
- *Address*:
- *Telephone*:
- *Telex*:
- *Price Category*:
- *Hardware Configuration*:
- *Expandibility*:
- *Installations in developing countries*:
- *System Description*:

B.2.31 Additional System:

- *System name*:
- *Vendor*:
- *Address*:
- *Telephone*:
- *Telex*:
- *Price Category*:
- *Hardware Configuration*:
- *Expandibility*:
- *Installations in developing countries*:
- *System Description*:

B.2.32 Additional System:

- *System name*:
- *Vendor*:
- *Address*:

Telephone:

Telex:

Price Category:

Hardware Configuration:

Expandibility:

Installations in developing countries:

System Description:

APPENDIX C
FULL TABLE OF CONTENTS

NO : 11409